

Приложение А. Глоссарий

Абстрактная операция – операция, которая объявляет сигнатуру, но не реализует ее. В C++ абстрактные операции соответствуют *исключительно виртуальным функциям-членам*.

Абстрактная связанность – говорят, что класс А *абстрактно связан* с абстрактным классом В, если в А есть ссылка на В. Такое отношение мы называем абстрактной связанностью, поскольку А ссылается на *тип* объекта, а не на конкретный объект.

Абстрактный класс – класс, единственным назначением которого является определение интерфейса. Абстрактный класс полностью или частично делегирует свою реализацию подклассам. Создавать экземпляры абстрактного класса нельзя.

Агрегированный объект – объект, составленный из подобъектов. Подобъекты называются *частями* агрегата, и агрегат отвечает за них.

Делегирование – механизм реализации, при котором объект перенаправляет или *делегировает* запрос другому объекту (уполномоченному). Уполномоченный выполняет запрос от имени исходного объекта.

Деструктор – в C++ это операция, которая автоматически вызывается для очистки объекта непосредственно перед его удалением.

Диаграмма взаимодействий – диаграмма, на которой показан поток запросов между объектами.

Диаграмма классов – диаграмма, на которой изображены классы, их внутренняя структура и операции, а также статические связи между ними.

Диаграмма объекта – диаграмма, на которой изображена структура конкретного объекта во время выполнения.

Динамическое связывание – ассоциация между запросом к объекту и одной из его операций, устанавливаемая во время выполнения. В C++ динамически связываться могут только виртуальные функции.

Дружественный класс – в C++: класс, обладающий теми же правами доступа к операциям и данным некоторого класса, что и сам этот класс.

Закрытое наследование – в C++: класс, наследуемый только ради реализации.

Замещение – переопределение операции, унаследованной от родительского класса, в подклассе.

Инкапсуляция – результат сокрытия представления и реализации в объекте. Представление невидимо и недоступно извне. Получить доступ к представлению объекта и модифицировать его можно только с помощью операций.

Инструментальная библиотека (toolkit) – набор классов, обеспечивающих полезную функциональность, но не определяющих дизайн приложения.

Интерфейс – набор всех сигнатур, определенных операциями объекта. Интерфейс описывает множество запросов, на которые может отвечать объект.

Каркас – набор взаимодействующих классов, описывающих повторно применимый дизайн некоторой категории программ. Каркас задает архитектуру приложения, разбивая его на отдельные классы с четко определенными функциями и взаимодействиями. Разработчик настраивает каркас под конкретное приложение путем порождения подклассов и составления композиций из объектов, принадлежащих классам каркаса.

Класс – определяет интерфейс и реализацию объекта. Описывает внутреннее представление и операции, которые объект может выполнять.

Композиция объектов – объединение нескольких объектов для получения более сложного поведения.

Конкретный класс – класс, в котором нет абстрактных операций. Может иметь экземпляры.

Конструктор – в C++: операция, автоматически вызываемая для инициализации новых экземпляров.

Метакласс – в Smalltalk классы являются объектами. Метакласс – это класс объекта-класса.

Наследование – отношение, которое определяет одну сущность в терминах другой. В случае *наследования класса* новый класс определяется в терминах одного или нескольких родительских классов. Новый класс наследует интерфейс и реализацию от своих родителей. Новый класс называется *подклассом* или *производным классом* (в C++). Наследование класса объединяет *наследование интерфейса* и *наследование реализации*. В случае наследования интерфейса новый интерфейс определяется в терминах одного или нескольких существующих. При наследовании реализации новая реализация определяется в терминах одной или нескольких существующих.

Объект – имеющаяся во время выполнения сущность, в которой хранятся данные и процедуры для работы с ними.

Операция – на данные объекта можно воздействовать только с помощью его операций. Объект выполняет операцию, когда получает запрос. В C++ операции называются *функциями-членами*, в Smalltalk – *методами*.

Операция класса – операция, определенная для класса в целом, а не для индивидуального объекта. В C++ операции класса называются *статическими функциями-членами*.

Отношение агрегирования – отношение агрегата и его частей. Класс определяет такое отношение для своих экземпляров, то есть агрегированных объектов.

Отношение осведомленности – говорят, что одному классу известно о другом, если первый ссылается на второй.

Параметризованный тип – тип, где некоторые составляющие типы оставлены неопределенными. Они передаются как параметры в точке использования. В C++ параметризованные типы называются шаблонами.

Паттерн проектирования – паттерн проектирования именуется, мотивирует и объясняет конкретный прием проектирования, который относится к задаче, часто возникающей при работе над объектно-ориентированными системами. Паттерн

описывает задачу, ее решение, область применимости этого решения и его результаты. Он также содержит рекомендации по реализации и примеры. Под решением понимается схема организации объектов и классов, позволяющая справиться с проблемой. Паттерн адаптируется для работы в конкретных условиях и реализуется в заданном контексте.

Переменная экземпляра – элемент данных, определяющий часть представления объекта. В C++ используется термин *данные-член*.

Подкласс – класс, наследующий другому классу. В C++ подкласс называется *производным классом*.

Подмешанный класс – класс, спроектированный так, чтобы сочетаться с другими классами путем наследования. Подмешанные классы обычно абстрактны.

Подсистема – независимая группа классов, функционирующих совместно для выполнения набора обязанностей.

Подтип – один тип называется подтипом другого, если интерфейс первого содержит интерфейс второго.

Полиморфизм – способность подставлять во время выполнения вместо одного объекта другой с совместимым интерфейсом.

Получатель – объект, которому направлен запрос.

Прозрачный ящик как способ повторного использования – стиль повторного использования, основанный на наследовании классов. Подкласс повторно использует интерфейс и реализацию родительского класса, но может также иметь доступ к закрытым для других аспектам своего родителя.

Протокол – расширяет концепцию интерфейса за счет включения допустимой последовательности запросов.

Родительский класс – класс, которому наследует другой класс. Синонимы – *суперкласс* (Smalltalk), *базовый класс* (C++) и *класс-предок*.

Связанность – степень зависимости компонентов программы друг от друга.

Сигнатура – под сигнатурой операции понимается сочетание ее имени, параметров и возвращаемого значения.

Ссылка на объект – значение, которое идентифицирует другой объект.

Супертип – тип родителя, которому наследует данный тип.

Тип – имя конкретного интерфейса.

Черный ящик как способ повторного использования – стиль повторного использования, основанный на композиции объектов. Объекты-компоненты не раскрывают друг другу деталей своего внутреннего устройства и потому могут быть уподоблены черным ящикам.