Перегрузка операторов позволяет определить действия, которые будет выполнять оператор. Перегрузка подразумевает создание функции, название которой содержит слово operator и символ перегружаемого оператора. Функция оператора может быть определена как член класса, либо вне класса. Перегрузить можно только те операторы, которые уже определены в C++. Создать новые операторы нельзя. Если функция оператора определена как отдельная функция и не является членом класса, то количество параметров такой функции совпадает с количеством операндов оператора. Например, у функции, которая представляет унарный оператор, будет один параметр, а у функции, которая представляет бинарный оператор, - два параметра. Если оператор принимает два операнда, то первый операнд передается первому параметру функции, а второй операнд - второму параметру. При этом как минимум один из параметров должен представлять тип класса

«Наблюдатель» (observer) – один из часто используемых шаблонов (паттернов) проектирования. Также известен как «подчинённые» (dependents) и «издатель-подписчик» (publisher-subscriber). Определяет зависимость типа «один ко многим» между объектами таким образом, что при возникновении некоторого события в одном из объектов, все объекты, подписавшиеся на оповещения об этом событии, извещаются о его возникновении. Шаблон «наблюдатель» позволяет организовать простое, прозрачное и логичное взаимодействие большого количества объектов различных типов и при этом, что очень важно, сделать зависимости между ними достаточно слабыми. Это обстоятельство позволяет избежать сильного связывания классов, что делает программу более гибкой, и разработчик получает возможность использовать в новых проектах ранее написанные классы без внесения в них существенных модификаций.

Паттерн Стратегия (Strategy) представляет шаблон проектирования, который определяет набор алгоритмов, инкапсулирует каждый из них и обеспечивает их взаимозаменяемость. В зависимости от ситуации мы можем легко заменить один используемый алгоритм другим. При этом замена алгоритма происходит независимо от объекта, который использует данный алгоритм.



