Метод Фабрика (Factory method)

Метод Фабрика е създаващ [шаблон за дизайн](http://bg.wikipedia.org/wiki/%D0%A8%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%B8_%D0%B7%D0%B0_%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD), който се използва в [обектно-ориентираното](http://bg.wikipedia.org/wiki/%D0%9E%D0%B1%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%B0%D0%BD%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5) [програмиране](http://bg.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5). Фабриката има за цел [инстанцирането](http://bg.wikipedia.org/w/index.php?title=%D0%98%D0%BD%D1%81%D1%82%D0%B0%D0%BD%D1%86%D0%B8%D1%8F&action=edit&redlink=1) на различни [обекти](http://bg.wikipedia.org/wiki/%D0%9E%D0%B1%D0%B5%D0%BA%D1%82), чиито типове не са [предефинирани](http://bg.wikipedia.org/wiki/%D0%94%D0%B5%D1%84%D0%B8%D0%BD%D0%B8%D1%86%D0%B8%D1%8F). Обектите се създават динамично в зависимост от [параметрите](http://bg.wikipedia.org/w/index.php?title=%D0%9F%D0%B0%D1%80%D0%B0%D0%BC%D0%B5%D1%82%D1%8A%D1%80&action=edit&redlink=1) предадени на фабриката. По принцип фабриките са единствени в дадена програма, затова за създаването им често се използва шаблонът [Сек (Singleton)](http://bg.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BA_(%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD)).

# Декоратор (Decorator)

Декоратор е структурен [шаблон за дизайн](http://bg.wikipedia.org/wiki/%D0%A8%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%B8_%D0%B7%D0%B0_%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD), който се използва в [обектно-ориентираното](http://bg.wikipedia.org/wiki/%D0%9E%D0%B1%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%B0%D0%BD%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5) [програмиране](http://bg.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5). Този шаблон може да бъде използван за разширяването на функционалността на опеределен [клас](http://bg.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5)) по времето на изпълнение на програмата, като запазва интерфейса му. Това разширение може да бъде постигнато чрез прибавянето на нов клас, който "обвива" оригиналния клас. Декораторите предоставят гъвкава алтернатива на наследяването за разширяване на функционалността.

# Стратегия (Strategy)

Стратегия е поведенчески [шаблон за дизайн](http://bg.wikipedia.org/wiki/%D0%A8%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%B8_%D0%B7%D0%B0_%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD), който се използва в [обектно-ориентираното](http://bg.wikipedia.org/wiki/%D0%9E%D0%B1%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%B0%D0%BD%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5) [програмиране](http://bg.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5). Шаблонът Стратегия се използва в случаите, когато е необходим динамичен избор на [алгоритъм](http://bg.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D1%8A%D0%BC). Дефинира се семейство от различни, но взаемозаменяеми алгоритми, всеки един от тях [капсулиран](http://bg.wikipedia.org/wiki/%D0%9A%D0%B0%D0%BF%D1%81%D1%83%D0%BB%D0%B8%D1%80%D0%B0%D0%BD%D0%B5) като отделен [обект](http://bg.wikipedia.org/wiki/%D0%9E%D0%B1%D0%B5%D0%BA%D1%82_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)). Стратегията позволява превключването от един към друг алгоритъм да става независимо от клиентския [код](http://bg.wikipedia.org/wiki/%D0%98%D0%B7%D1%85%D0%BE%D0%B4%D0%B5%D0%BD_%D0%BA%D0%BE%D0%B4), който ги използва.