**Singleton**

Това е най-популярният и използван шаблон. Позволява на определен клас да има само една инстанция и дефинира откъде да се вземе тази инстанция. Това е полезно, когато точно един обект се нуждае да координира действия през системата. Типични примери са класове, които дефинират връзка към единствени неща (виртуалната машина, операционна система, мениджър на прозорците при графично приложение, файлова система), както и класовете от следващия шаблон (factory).

Шаблонът може да претърпи много оптимизации, например т.нар. "мързеливо инициализиране" (lazy init) на единствената променлива за спестяване на памет, но това е класическата му форма.

**Factory Method**

Factory method е друг много разпространен шаблон. Той е предназначен да "произвежда" обекти. Инстанцирането на определен обект не се извършва директно, а се прави от factory метода. Това позволява на factory метода да реши коя конкретна инстанция да създаде. Решението може да зависи от външната среда, от параметър или от някаква системна настройка. Factory методите капсулират създаването на обекти. Това е полезно, ако процесът по създаването е много сложен – например зависи от настройки в конфигурационните файлове или от данни въведени от потребителя. По принцип фабриките са единствени в дадена програма, затова за създаването им често се използва Singleton шаблона.

**Abstract Factory**

Фабриката е средство за създаване на обекти. Целта на този шаблон за дизайн е да изолира създаването на обектите от тяхното използване. Абстрактната фабрика капсулира група от методи имащи близко предназначение. Клиентският код създава конкретна имплементация на абстрактната фабрика, след това използва основния интерфейс, за да създава конкренти обекти. Клиентът не е задължен да знае коя от тези фабрики е създала конкретния обект, защото той използва само основния интерфейс към създадените обекти. Този шаблон позволява замяната на конкретни класове, дори по време на изпълнение, без да е нужна промяна на кода, който ги използва.