## Spring

Jaká je filozofie Springu? Nejdůležitějším slovem v souvislosti se Springem je: kontext. Jádro Springu je postaveno na využití návrhového vzoru [*Inversion of Control*](http://cs.wikipedia.org/w/index.php?title=Inversion_of_Control&action=edit&redlink=1) a je označován jako IoC kontejner. Tento návrhový vzor funguje na principu přesunutí zodpovědnosti za vytvoření a provázání objektů z aplikace na framework. Objekty lze získat prostřednictvím [*vkladání závislostí*](http://cs.wikipedia.org/w/index.php?title=Dependency_Injection&action=edit&redlink=1) *(dependency injection)*, což je speciální případ Inversion of Control. Dependency Injection řeší vlastní způsob vložení objektů. Základní tři způsoby vložení objektů jsou Setter Injection, Constructor Injection a Interface Injection. Objekty vytvořené kontejnerem jsou nazývány [*Beans*](http://cs.wikipedia.org/wiki/Java_Bean). Objekty jsou frameworkem vytvořeny typicky na základě načtení konfiguračního souboru ve formátu XML, který obsahuje definice těchto Beans.[3]

Jinými slovy, Spring Framework je založen zhruba na následovné filozofii: Mezi objekty jsou různé vztahy (*dependencies,* tj. odkazy z jedné instance na instanci jinou). Pojďme tento model zkonstruovat tak, že nejdříve vytvoříme všechny potřebné objekty (bez vzájemných vztahů) a pak v druhém kroku vytvořme tyto vztahy. Nebo ještě lépe: vytvořme tyto vztahy až v okamžiku, když je budeme potřebovat. Vytváření vztahů se v Spring hantýrce říká *wiring*.

(Normálně je to tak, že jednotlivé instance, pokud obsahují odkazy na jiné, tak si je obvykle sami vytvoří, např. v rámci konstruktoru. Vnitřní instance je tak „uzavřená“ ve vnější instanci a není k ní zvenčí přístup, což může mít řadu nevýhod).

Spring Framework se nezabývá řešením již vyřešených problémů. Místo toho využívá prověřených a dobře fungujících existujících open-source nástrojů, které v sobě integruje. Tím se stává jejich použití často jednodušším.[3]