



Factory method

Virtual constructor,
Továrenská metóda

Definuje rozhranie pre vytváranie objektov, avšak necháva na potomkoch, aká trieda sa má vytvoriť. Továrenská metóda umožňuje triede delegovať vytváranie inštancií na potomkov.

Define an interface for creating an object, but let subclasses decide which class to instantiate. Factory method lets a class defer instantiation to subclasses.

Návrhové vzory



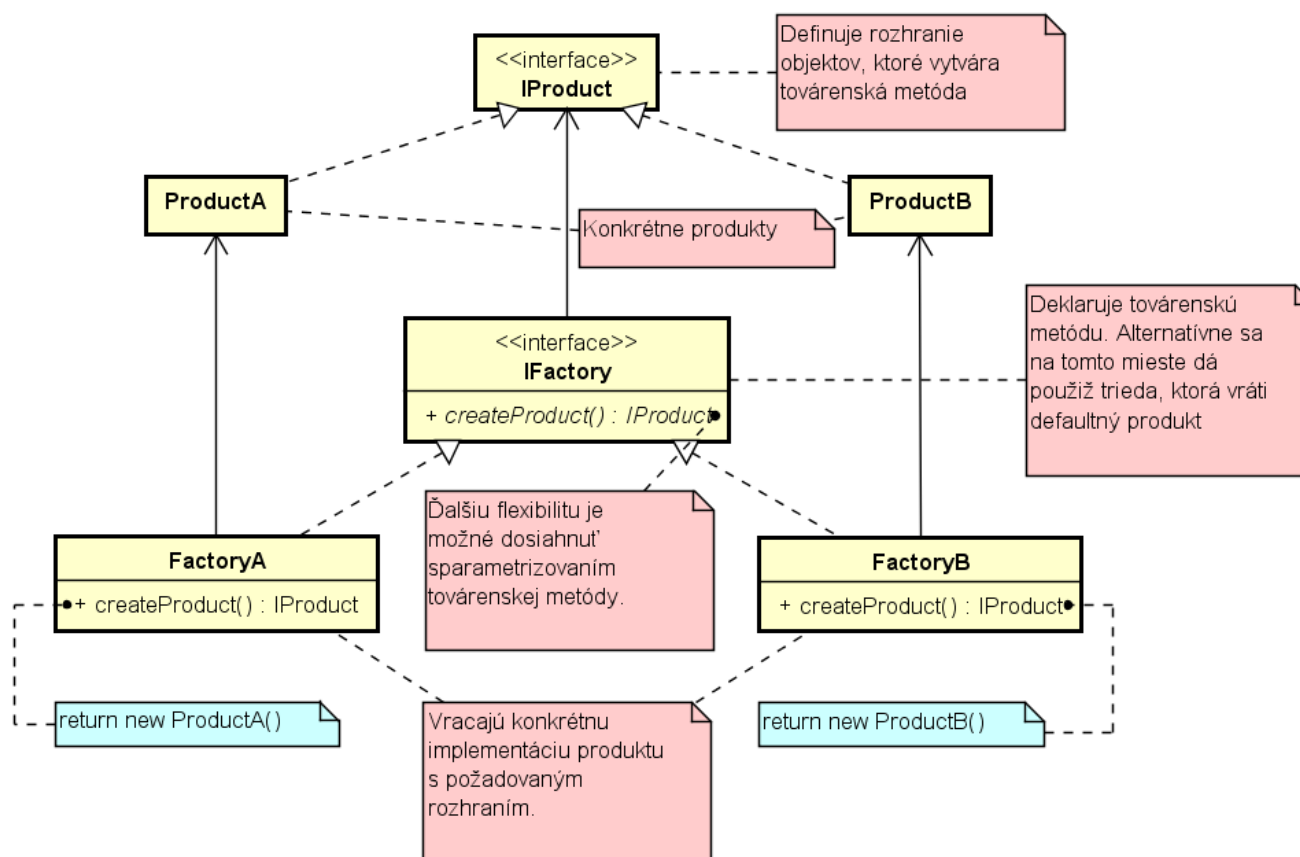
Motivácia

- Typickým príkladom továrenskej metódy sú iterátory kontajnerov:
 - Existuje všeobecný zoznam, ktorý deklaruje, že poskytuje všeobecný iterátor.
 - To, aký iterátor sa vráti, určí potomok – iterátor zreťazeného zoznamu je iný ako iterátor ArrayList-u – oba iterátory však majú požadované rozhranie.

Aplikovateľnosť

- Továrenskú metódu je dobré použiť, keď:
 - trieda nevie dopredu, akých konkrétnych potomkov má vytvárať.
 - trieda požaduje, aby potomok špecifikoval objekty, ktoré sú vytvárané.
 - trieda deleguje časť svojich zodpovedností do svojho potomka a pritom je požadované centralizovať znalosti o tom, ktorý potomok je týmto delegátom.

Implementácia



powered by Astah

Dôsledky

- Odstraňuje potrebu uvedenia konkrétnej triedy pri vzniku objektu, pracuje iba s rozhraním `IProduct`, a to úplne postačuje na prácu s akýmikoľvek objektami, ktoré ho implementujú.
- Nevýhodou je, že sa môže javiť ako nepohodlná, pretože pre vytvorenie špecifického objektu s rozhraním `IProduct` je potrebné vytvoriť nový objekt implementujúci rozhranie `IFactory` – tu môže nastať problém ak sa namiesto rozhrania použije dedenie a tým sa predpíše hierarchia.
- Podporuje princíp programovať proti rozhraniu – pracuje sa iba so všeobecnými rozhraniami.
- Vytváranie objektov pomocou továrenskej metódy je pružnejšie ako priame vytváranie objektov.



Príbuzné vzory

- *Abstract factory* – často sa na jej implementáciu využívajú továrenské metódy.
- *Template method* - tieto vzory často stavajú na továrenských metódach.
- *Prototype* – rieši podobné problémy, nevyžaduje zdediť továreň, avšak predpisuje klonovaciu metódu.

Upozornenie

- Tieto študijné materiály sú určené výhradne pre študentov predmetu 5I132 Návrhové vzory (Design Patterns) na Fakulte riadenia a informatiky Žilinskej univerzity v Žiline.
- Reprodukovanie, šírenie (i častí) materiálov bez písomného súhlasu autora nie je dovolené.

Ing. Michal Varga, PhD.
Katedra informatiky
Fakulta riadenia a informatiky
Žilinská univerzita v Žiline
Michal.Varga@fri.uniza.sk

