

8 Návrhové vzory I

Wednesday, 19 January 2022 09:02

Návrhové vzory pro vytváření instancí, rozšiřování funkcionality.

- návrhový vzor
 - šablona pro řešení jednoho nebo více konkrétních problémů
 - použití pro zjednodušení problémů
 - relativně nízká chybovost díky ověřené funkčnosti
 - a. vzory chování
 - mění nebo vytváří chování objektů
 - MVC, interpreter, observer, iterátor
 - b. strukturální vzory
 - řeší strukturu systému a jeho komponent
 - zástupce, adaptér, prázdný a neměnný objekt
 - c. vzory pro vytváření
 - řeší problémy v souvislosti s tvorbou objektů
 - singleton, pool, knihovna, tovární metoda
1. vzory pro vytváření
- a. **factory method** (tovární metoda)
 - vytvoření nové instance třídy na základě parametrů
 - využití, kde jsou vytvářené objekty odvozeny od stejné třídy
 - často u immutable (neměnných) objektů, jejichž stav nelze po vytvoření změnit
 - b. **singleton** (jedináček)
 - maximálně jedna instance dané třídy
 - tato instance by měla být centralizovaně dostupná
 - nesmí být umožněno vytvářet instance mimo samotný singleton
 - musí být thread-safe a serializovatelný
 - c. **pool** (fond)
 - umožňuje znovu použít existující instance třídy
 - řešení paměťově náročných tříd nebo tříd s velkým množstvím instancí
 - implementace pomocí kolekce, která bude obsahovat všechny instance
 - d. **utility** (knihovna)
 - skupina metod, které spolu nějak souvisí a jsou používány společně
 - metody uloženy v jedné třídě, aby z nich nešlo dědit a vytvářet instance → static
 - e. **originál**
 - třída, jejíž instance jsou svými parametry unikátní
 - neexistuje více jak jedna instance se stejnými parametry
2. strukturální vzory
- a. **flyweight** (muší váha)
 - podobný vzoru fond, ale místo instancí pracuje s prostředky, se kterými instance pracují
 - řešení paměťově náročných tříd nebo tříd s velkým množstvím instancí
 - b. **immutable** (neměnné) objekty
 - hodnotový objekt, u kterého nejde změnit hodnotu
 - zvyšuje bezpečnost programu
 - při modifikaci hodnoty se vytváří nový objekt a odkaz se přemísťuje na něj
 - starý objekt zůstává bez odkazu a je odstraněn garbage collectorem
 - např. string
 - mutable ekvivalentem stringu je StringBuilder
 - c. **crate** (přepravka)
 - předávání několika samostatných informací
 - atributy se mohou definovat jako konstantní, aby byla zajištěna neměnnost
 - např. souřadnice

3. vzory chování

a. *servant* (služebník)

- je-li potřeba, aby instance více tříd, které nemohou mít stejného předchůdce, obsahovaly stejnou funkčnost
- dvě možnosti:
 - klient služebníka zná
 - klient služebníka nezná

b. *null object* (prázdný objekt)

- použití tam, kde se nehodí čistý null

c. *template method* (šablonová metoda)

- definuje kostru algoritmu
- potomci mohou předefinovat určité kroky algoritmu bez zásahu do kostry
- rodičovská třída deklaruje placeholder algoritmu a potomci je implementují