

8

Teoretické východiská

Návrhové vzory

Návrhové vzory

- Architektúra systému:
 - Menšia
 - Jednoduchšia
 - Pochopiteľnejšia
- Skúsenosti – opakujúce sa vzory tried a objektov
- Dobrý objektový návrh a návrhové vzory

Návrhový vzor

- Systematicky
 - Pomenováva, vysvetľuje, vyhodnocuje opakujúce sa návrhy
- Zjednodušujú ich opätovné použitie
- Pomáha pri ich výbere
- Vylepšujú dokumentáciu
- Vylepšujú údržbu existujúcich systémov

Zloženie návrhových vzorov

- Názov vzoru
- Problém
- Riešenie
- Dôsledky

Popis návrhových vzorov

- Názov vzoru a klasifikácia
- Zmysel
- Alias
- Motív
- Aplikovateľnosť
- Štruktúra
- Účastníci vzoru
- Spolupráca
- Dôsledky
- Implementácia
- Príklad vzoru
- Súvisiace vzory

Singleton: klasifikácia a zmysel, alias

- Object
- Creational
- Zabezpečuje, že trieda má len jednu inštanciu a poskytuje k nej globálny prístupový bod

Katalóg vzorov

- Podľa účelu
 - Pre tvorbu objektov
 - Pre štruktúry
 - Riešiacie chovanie
- Podľa rozsahu
 - S rozsahom platnosti tried
 - S rozsahom platnosti objektov

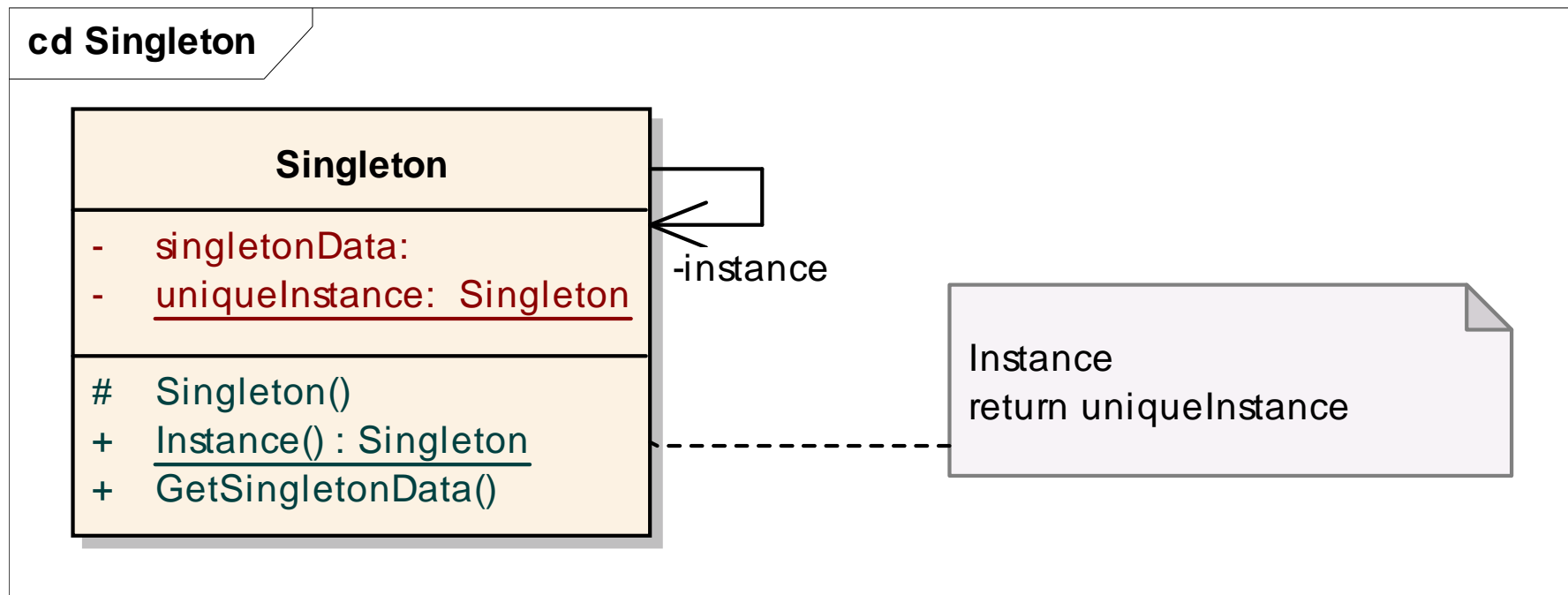
Singleton: Motív

- Pre niektoré triedy je dôležité mať presne jednu inštanciu
- Ako zabezpečiť, že trieda má len jednu inštanciu a tá je ľahko prístupná?

Singleton: Aplikovateľnosť

- Je požadovaná práve jedna inštancia triedy a musí byť prístupná klientom zo známeho prístupového bodu
- Keď základná inštancia by mala byť rozširovaná dedením a klienti by mali byť schopný používať rozšírenú inštanciu bez modifikácie vlastného kódu

Singleton: Štruktúra



Singleton: Účastníci a spolupráca

- Singleton
 - Definuje operáciu Instance, ktorá umožňuje klientom prístup k jedinej inštancii. Instance je operácia triedy (v C++ static).
 - Môže byť zodpovedný za vytvorenie vlastnej jedinej inštancie
- Klienti pristupujú k inštancii triedy Singleton výhradne cez Instance operáciu

Singleton: Dôsledky

- Kontrolovaný prístup k jedinej inštancii
- Redukuje menný priestor
- Umožňuje redefiníciu operácií a reprezentácie
- Umožňuje meniteľné množstvo inštancií
- Viacej flexibilné ako operácie tried

Singleton: Implementácia

- Zabezpečenie jedinej inštancie
- Vytváranie potomkov Singletonu

Singleton: Příklad vzoru

```
class Singleton {  
private:  
    static Singleton* uniqueInstance;  
protected:  
    Singleton(){};  
public:  
    static Singleton* Instance() {  
        if (uniqueInstance == null) {  
            uniqueInstance = new Singleton;  
        }  
        return uniqueInstance;  
    };  
}  
  
Singleton* Singleton::uniqueInstance = 0;
```

Súvisiace vzory

- Abstract Factory
- Builder
- Prototype

Použitie návrhových vzorov

- Preštudovať si vzor
- Vrátiť sa k štruktúre, účastníkom a spolupráce
- Pozrite si príklad
- Vyberte si mená tried
- Definujte triedy
- Vyberte názvy pre metódy
- Implementujte metódy

Princípy

Objektovo orientovaný návrh

- Treba zohľadniť
 - Zapúzdrenie
 - Granularita
 - Závislosť
 - Flexibilitu
 - Výkonnosť
 - Re-use
 - ...

Flexibilita návrhu

- Závislosť objektov na triede
- Závislosť na špecifickej operácii
- Závislosť na SW a HW
- Závislosť na objektovej implementácii
- Závislosť na algoritmu
- Tesné väzby objektov
- Preceňovanie dedenia
- Zmeny bez prístupu ku kódu triedy

Vonkajší pohľad na objekt

- Operácia – správa;
- Rozhranie objektu – interface
- Typ, subtyp
- Neskorá väzba a polymorfizmus

Vnútorný pohľad na objekt

- Trieda
- Inštancia
- Dedičnosť, rodič, potomok,
- Abstraktná trieda
- Prekrývanie metód

1. Princíp

Programovanie „do“ rozhrania,
nie implementácie

Trieda a interface

- Dedenie z triedy a rozhrania
- Programovanie „do“ rozhrania, nie implementácie
 - Klienti sú nezávislí na špecifickom type objektu
 - Klienti sú nezávislí na triedach, ktoré implementujú tieto objekty

2. princíp

Uprednostňovanie kompozície
pred dedičnosťou tried

Dedičnosť a kompozícia

Definovaná staticky

- + Priamočiara

- + Modifikácie

- Zmeny implementácie za behu

- Definícia potomkov v predkovi

Definovaná dynamicky

- + Zmeny implementácie za behu

- + Implementačná závislosť

- + Menšie triedy a hierarchie tried

- Viacej objektov

- Závislosť na vzťahoch objektov