Metaprogramování

Multiparadigmatický přístup v softwarovém inženýrství

Kyjev, 2015 — 2022

Anotace

Všechny programy jsou data. Některá data jsou interpretována jako hodnoty, jiná jako typy těchto hodnot a další jako instrukce pro zpracování prvních dvou. Jakákoli paradigmata a programovací techniky jsou jen způsobem, jak vytvořit metadata, která dávají pravidla a posloupnost toku zpracování jiných dat. Multiparadigmatické programování bere to nejlepší ze všech paradigmat a sestavuje z něj syntaktické konstrukce, které umožňují popsat předmětnou oblast jasněji a pohodlněji. Vysokoúrovňové DSL (doménové jazyky) promítáme do nízkoúrovňových strojových instrukcí prostřednictvím mnoha vrstev abstrakcí. Zde je důležité reprezentovat úkol co nejúčinnějším způsobem pro zpracování na úrovni stroje, nikoli fanaticky následovat jedno paradigma. Nejúčinnější je ten, který s menším počtem vrstev a závislostí, nejlépe čitelný, udržovatelný a upravitelný pro člověka, který zajišťuje spolehlivost kódu a testovatelnost, rozšiřitelnost, opětovnou použitelnost, jasnost a flexibilitu konstrukcí metadat na každé úrovni. Věříme, že takový přístup nám umožní získat jak rychlé první výsledky ve vývoji, tak i neztrácet výkon při velkém toku změn ve fázích, kdy projekt již dosáhl vysoké zralosti a složitosti. Pokusíme se zvážit techniky a principy programování z různých paradigmat prizmatem metaprogramování a změnit tím, ne-li samotné softwarové inženýrství, ale alespoň jeho chápání novými generacemi inženýrů.

Obsah

8.5. Memoizace 8.6. Návrhové vzory: Factory, Poll 8.7. Typovaná pole 8.8. Projekce 8.9. I/O (vstup-výstup) a soubory

9. Architektura 9.1. Dekompozice, pojmenování a spojování 9.2. Interakce mezi softwarovými komponentami 9.3. Propojení přes jmenné prostory 9.4. Interakce s voláními a zpětnými voláními 9.5. Interakce s událostmi a zprávami 9.6. Rozhraní, protokoly a smlouvy 6.7. Cibulová (onion) nebo vícevrstvá architektura

10. Základy paralelních výpočtů 10.1. Asynchronní programování 10.2. Paralelní programování, sdílená paměť a synchronizační primitiva 10.3. Asynchronní primitiva: Thenable, Promise, Future, Deferred 10.4. Koprogramy, gorutiny, async/await 10.5. Adaptéry mezi asynchronními kontrakty 10.6. Asynchronní a paralelní kompatibilita 10.7. Přístup předávání zpráv a model aktorů 10.8. Asynchronní fronty i asynchronní kolekce 10.8. Bezzámkové datové struktury (lock-free)

11. Pokročilá programovací paradigmata 11.1. Generické programování 11.2. Událostní a reaktivní programování 11.3. Programování automatů: konečné automaty (stavové stroje) 11.4. Jazykově orientované programování a DSL 11.5. Programování toku dat 11.6. Metaprogramování 11.7. Dynamická interpretace metamodelu

12. Databáze a perzistentní úložiště 12.1. Historie databáze a navigační databáze 12.2. Klíč-hodnota a další abstraktní datové struktury 12.3. Relační datový model a ER-diagramy 12.4. Bezschémové, objektově orientované a dokumentově orientované databáze 12.5. Hierarchické a grafové databáze 12.6. Sloupcové databáze a databáze v paměti 12.7. Distribuované databáze

13. Distribuované systémy 13.1. Meziprocesová komunikace 13.2. Bezkonfliktní replikované datové typy (CRDT) 13.3. Konzistence, dostupnost a distribuce 13.4. Strategie řešení konfliktů 13.5. Konsensuální protokoly 13.6. CQRS, EventSourcing

Úvod

Neustálé přehodnocování své činnosti, i té nejjednodušší, by mělo provázet inženýra celý život. Zvyk zapisovat si své myšlenky slovy a zdokonalovat své formulace v tom hodně pomáhá. Tento text se objevil jako moje fragmentární poznámky, psané v různých letech, které jsem shromažďoval a mnohokrát kriticky korigoval. Já jsem často nesouhlasil sám se sebou, když jsem si znovu četl pasáž poté, co chvíli ležela. Proto jsem na textu pracoval tak dlouho, dokud jsem sám nesouhlasil s tím, co bylo napsáno po dlouhé době držení materiálu. Bylo mým úkolem psát co nejstručněji a velké fragmenty jsem opakovaně přepisoval, když jsem zjistil, že by se daly vyjádřit stručněji. Struktura textu a obsah se začaly objevovat po prvním roce výuky, ale v desátém ročníku jsem se rozhodl šířit materiály nejen ve formě otevřených videopřednášek, jak jsem to dělal už asi pět let, ale i ve formě textu. To umožnilo všem ze společenství Metarhia podílet se na tvorbě knihy, díky čtenářům rychle objevovat překlepy a nepřesnosti, a také pro mnohé je formát knihy prostě pohodlnější. Aktuální verzi naleznete na https:// github.com/ HowProgrammingWorks/ Book, bude neustále aktualizována. Žádosti o opravy a doplnění směřujte v angličtině na [issues] (https:// github.com/ HowProgrammingWorks/ Book/ issues), nové nápady zasílejte v libovolném jazyce na [discussions] (https:// github.com/ HowProgrammingWorks/ Book/ discussions), vaše vlastní doplňky a opravy by měly být provedeny formou pull-request do repositáře knihy.