

Refaktorování (Čištění kódu)

Refaktorování je nezbytnou součástí životního cyklu softwaru. Stejně jako zahrada vyžaduje pravidelné pletí, aby nezarostla plevelem, kód vyžaduje refaktorování, aby se v něm nehromadil tzv. **technický dluh**.

Základní pravidlo zní: **Pokud kód po refaktorování dělá něco jiného než předtím, nejednalo se o refaktorování, ale o opravu chyb nebo vývoj nové funkce.**

Hlavní cíle refaktorování

- **Zlepšení čitelnosti:** Aby kód pochopil i někdo jiný (nebo autor sám po půl roce).
 - **Odstranění duplicit:** Důsledné uplatňování principu **DRY**.
 - **Snížení složitosti:** Rozbití obřích funkcí na menší, přehlednější celky.
 - **Usnadnění budoucích změn:** Čistý kód se mnohem lépe rozšiřuje o nové funkce.
-

Kdy refaktorovat?

Nejlepším průvodcem jsou tzv. **Code Smells** (zápachy kódu) - náznaky, že s kódem není něco v pořádku:

- **Duplicitní kód:** Stejná logika na více místech.
 - **Příliš dlouhá metoda:** Funkce, která má stovky řádků a dělá deset věcí najednou.
 - **Obří třída (God Object):** **Třída**, která se snaží ovládat celý program.
 - **Nesrozumitelná jména:** Proměnné jako `a`, `temp` nebo `data123`.
-

Běžné techniky refaktorování

1. Extraxe metody (Extract Method)

Vezmete část složitého kódu z dlouhé funkce, přesunete ji do nové, samostatné funkce a tu původní nahradíte voláním této nové funkce.

2. Přejmenování (Rename)

Změna názvů proměnných, metod nebo tříd tak, aby přesně vyjadřovaly svůj účel. Moderní vývojová prostředí (IDE) toto zvládnou automaticky v celém projektu.

3. Nahrazení magických čísel konstantami

Místo čísla 86400 v kódu použijete pojmenovanou konstantu SEKUND_ZA_DEN.

Postup při bezpečném refaktorování

Refaktorování může být riskantní, proto by mělo probíhat v malých krocích:

- Mějte testy:** Před začátkem byste měli mít sadu automatických testů, které ověří, že kód funguje správně.
- Malé změny:** Udělejte jednu drobnou úpravu (např. přejmenování).
- Spusťte testy:** Pokud všechny projdou, pokračujte. Pokud ne, vraťte změnu zpět.
- Čistěte:** Opakujte, dokud kód není "voňavý".

Rozdíl: Refaktorování vs. Optimalizace

Činnost	Co se mění?	Cíl
Refaktorování	Vnitřní struktura.	Čitelnost a udržovatelnost.
Optimalizace	Využití zdrojů.	Vyšší rychlost, nižší spotřeba paměti.

Pozor: Optimalizovaný kód je často hůře čitelný. Proto by se mělo nejprve refaktorovat (pro čistotu) a teprve v případě nutnosti optimalizovat (pro výkon).

Související pojmy: DRY princip, Technický dluh, Jednotkové testy (Unit Testing), SOLID principy, KISS.

From:
<https://serviceit.cz/> - IT ENCYKLOPEDIE

Permanent link:
<https://serviceit.cz/doku.php?id=refaktorovani>

Last update: 2025/12/31 20:09



