

Dobre praktyki programistyczne

Jacek Piwoński

Listopad, 2019

Idea

- Większość czasu pracy programisty stanowi obcowanie z już istniejącym kodem
- Raz dobrze napisany kod odwdzięcza się za każdym razem, gdy ktoś z niego korzysta
- Czysty kod to oszczędność czasu

Zasada skautów

- Nawet dobrze napisany kod degraduje się z czasem
- W momencie obcowania z kodem należy zastanowić się, czy można w nim coś poprawić lub zaktualizować
- Zostawianie kodu czystsze niż zastany przeciwdziała jego degradacji

Znaczące nazwy

- Znaczące nazwy są podstawą czytelności kodu
- Dobra nazwa pozwala poznać działanie danego elementu bez zagłębiania się w jego implementację
- Prawidłowa nazwa zapobiega dezinformacji

Cechy dobrej nazwy

Jaka jest dobra nazwa?

- Przedstawia prawdziwe intencje
- Umożliwia łatwe wyszukanie w projekcie
- Jest spójna w obrębie całego projektu
- Unika kodowania
- Da się ją wymówić (wyjątkiem są niektóre skróty ważne biznesowo)

Dobra funkcja

Podstawowe założenia dobrej funkcji

- Jest mała
- Ma jedno zadanie i tylko je wykonuje
- Ma jeden poziom abstrakcji

Ponadto

- Nazwa jasno przedstawia intencje i skutki funkcji
- Unika efektów ubocznych
- Minimalizuje liczbę argumentów

Zalety

Co daje tak napisana funkcja?

- Ułatwia zrozumienie danego elementu systemu, co przekłada się na oszczędności w czasie
- Ułatwia zmiany
- Ułatwia testowanie poszczególnych funkcjonalności

Czysta klasa

- Jest mała
- Posiada możliwie mało odpowiedzialności (SRP)
- Posiada spójną kolejność poszczególnych elementów w skali projektu

Idea

- Komentarz nie służy do upiększania złego kodu
- Dobry kod sam tłumaczy swoje działanie

Dobre komentarze

- Komentarze prawne
- Wyjaśnienie zamierzeń programisty
- Ostrzeżenia o konsekwencjach
- TODO
- Javadoc

Bibliografia

- Robert C. Martin - Czysty kod
- Victor Rentea - The Art of Clean Code