

Notatki Czysty Kod - Uncle Bob

1. Używaj nazw przedstawiających intencje
2. Unikaj dezinformacji
3. Twórz wyraźne różnice (nazywanie zmiennych)
4. Twórz nazwy, które da się wymówić
5. Korzystaj z nazw istnych do wyszukiwania
6. Klasy i funkcje powinny być na tyle ~~dużo~~ małe, aby nie były potrzebne
7. Nie dodajemy przedrostków do interfejsów
8. Nazwy klas powinny być wyrażeniami rzeczownikowymi
9. Nazwy metod powinny być wyrażeniami czasownikowymi
10. Unikamy jednego słowa do dwóch zagadnień
11. Nie piszemy długich funkcji
12. Max dwa wcięcia w funkcji
13. Funkcje powinny wykonywać jedną operację
14. Jeden poziom abstrakcji w funkcji
15. Nie przekraczamy dwóch argumentów w funkcjach
16. Rozdzielaj polecenie i zapytanie
17. Stosuj wyjątki zamiast zwrocenie kodów błędów
18. Wyodrębnij bloki try-catch
19. Komentarze nie są szminką dla kodu. Czytelny kod nie wymaga komentarzy
20. Komentarzujemy tylko wyjaśnienie, zamiary i ostrzeżenia
21. Nie komentuj kodu
22. Zmienne powinny być deklarowane tak blisko użycia jak tylko to możliwe

23. Funkcje zależne mają znajdować obok siebie
24. Koligacja koncepcyjna
25. Klasa nie powinna przepychać zmiennych przez gettery i settery. Powinna udostępnić interfejs pozwalający użytkownikom na manipulowanie istotą danych bez konieczności znajomości jej implementacji.
26. Rozpoczynamy od zapisania try-catch-finally
27. Definiujemy klasy wyjątków w zależności od potrzeb wywołującego
28. Nie zwracamy null.
29. Nie przekazujemy null.
30. Jedna seria na test.
31. Jedna koncepcja na test.
32. Wygląd klasy

Lista zmiennych → publiczne statyczne statyczne →

→ prywatne statyczne statyczne → prywatne zmienne instancyjne

3.3 Systemy oprogramowania powinny oddzielić proces uruchomienia w którym są budowane obiekty aplikacji i łączone ze sobą zależności, od logiki działania, która jest wykorzystywana po uruchomieniu.

34. Optymalna architektura systemu składa się z modularnych domen ~~problemów~~ problemów; każda z nich jest implementowana z użyciem obiektów, np. POJO (Plain Old Java Object). Osobne domeny są integrowane ze sobą przy użyciu minimalnie inwazyjnych narzędzi aspektowych lub podobnych.