

- Projekt jest „prosty”, goly

- przechodzi wszystkie testy
- nie zawiera powtórzeń
- wyraża intencje programisty
- minimalizuje liczbę klas i metod

- Posiadanie testów eliminuje obawę, że proces czyszczenia kodu uszkodzi go.

- Single Responsibility Principle

Zasada mówiąca, że nie powinno być więcej niż jednego powodu do zmiany klasy.

- Open / Closed Principle

Jedna z zasad programowania mówiąca, że elementy systemu takie jak klasy, funkcje, moduły powinny być otwarte na rozszerzenie, zamknięte na modyfikację.

- Prawo Demeter (Principle of least knowledge)

Metoda danego obiektu może odwoływać się jedynie do metod należących do

- tego samego obiektu
- dowolnego parametru przekazanego do niej
- dowolnego obiektu stworzonego przez nią
- dowolnego składnika klasy, do której należy dana metoda

- Trzy prawa TDD

① Nie można zacząć pisać kodu produkcyjnego do momentu napisania niespełnionego testu jednostkowego

② Nie można napisać więcej testów jednostkowych, które są wystarczające do niespełnienia testu, a brak kompilacji jest jednocześnie niedanym testem

③ Nie można pisać większej ilości kodu ~~produkcyjnego~~ produkcyjnego, niż wystarczy do spełnienia obecnie niespełnionego testu