

Pomiar jakości oprogramowania

Cel pomiarowy: monitorowanie poprawności działania zaimplementowanych transformacji

Definicja:

Dokonać analizy testów

W celu scharakteryzowania (oceny)

W odniesieniu do poprawności transformacji

Z perspektywy programisty

W kontekście deweloperów aplikacji

Pytanie: Jakie jest pokrycie transformacji testami?

Miara: Groovy test with coverage

Obiekt pomiaru: produkt

Sposób pomiaru: obiektywny

Rodzaj miary: bezpośrednia

Skala pomiaru: stosunkowa

Element	Class, %	Method, %	Line, %
ExpandDecorator	100% (1/1)	100% (3/3)	100% (29/29)
LatexDecorator	100% (1/1)	100% (3/3)	100% (11/11)
LetterTransformerDecorator	100% (1/1)	100% (6/6)	85% (53/62)
NumToTextDecorator	100% (1/1)	100% (9/9)	84% (45/53)
RepetitionDeleteDecorator	100% (1/1)	100% (3/3)	100% (14/14)
Response	0% (0/1)	0% (0/5)	0% (0/8)
ShrinkDecorator	100% (1/1)	100% (3/3)	100% (26/26)
TextTransformer	100% (1/1)	100% (1/1)	100% (2/2)
TransformerDecorator	100% (1/1)	50% (1/2)	75% (3/4)
TransformerService	100% (1/1)	100% (1/1)	100% (9/9)
TransformRequestModel	100% (2/2)	83% (10/12)	83% (10/12)

Cel pomiarowy: monitorowanie łatwości utrzymania aplikacji

Definicja:

Dokonać analizy kodu

W celu scharakteryzowania (oceny)

W odniesieniu do łatwości utrzymania

Z perspektywy programisty

W kontekście deweloperów aplikacji

Pytanie: Jaka jest poprawność składni kodu?

Miara: IntelliJ Inspector

Obiekt pomiaru: produkt

Sposób pomiaru: obiektywny

Rodzaj miary: bezpośrednia

Skala pomiaru: absolutna

▼ General 5 errors
▼ Annotator 2 errors
▼ script.js 2 errors
Duplicate declaration
Duplicate declaration
▼ Syntax error 3 errors
▼ API.md 3 errors
Expecting newline or semicolon
, expected
, expected