

Mutation testing report

Killed 14 out of 19 mutants

File	Total	Killed	% killed	Survived
com/automationpanda/example/calc_class.py	4	4	100.00	0
com/automationpanda/example/calc_func.py	10	15	66.67	5

com/automationpanda/example/calc_func.py

Killed 10 out of 15 mutants

Survived

Survived mutation testing. These mutants show holes in your test suite.

Mutant 8

```
--- com/automationpanda/example/calc_func.py
+++ com/automationpanda/example/calc_func.py
@@ -17,7 +17,7 @@

def divide(a, b):
    # automatically raises ZeroDivisionError
-   return a * 1.0 / b
+   return a / 1.0 / b

def maximum(a, b):
```

Mutant 11

```
--- com/automationpanda/example/calc_func.py
+++ com/automationpanda/example/calc_func.py
@@ -21,7 +21,7 @@

def maximum(a, b):
-   return a if a >= b else b
+   return a if a > b else b

def minimum(a, b):
```

Mutant 12

```
--- com/automationpanda/example/calc_func.py
+++ com/automationpanda/example/calc_func.py
@@ -25,7 +25,7 @@

def minimum(a, b):
-   return a if a <= b else b
+   return a if a < b else b

def logarithm(a, base=10):
```

Mutant 16

```
--- com/automationpanda/example/calc_func.py
+++ com/automationpanda/example/calc_func.py
@@ -34,7 +34,7 @@
    Levanta ValueError se 'a' <= 0 ou base <= 1.
    """
    if a <= 0:
-       raise ValueError("O logaritmo só é definido para números positivos.")
+       raise ValueError("XXO logaritmo só é definido para números positivos.XX")
    if base <= 1:
```

```
        raise ValueError("A base do logaritmo deve ser maior que 1.")
    return math.log(a, base)
```

Mutant 19

```
--- com/automationpanda/example/calc_func.py
+++ com/automationpanda/example/calc_func.py
@@ -36,5 +36,5 @@
     if a <= 0:
         raise ValueError("O logaritmo só é definido para números positivos.")
     if base <= 1:
-        raise ValueError("A base do logaritmo deve ser maior que 1.")
+        raise ValueError("XXA base do logaritmo deve ser maior que 1.XX")
     return math.log(a, base)
```