

Mutation testing report

Killed 10 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	6	15	40.00	9

com/automationpanda/example/calc_func.py

Killed 6 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

import math
```

Mutant 13

```
--- com/automationpanda/example/calc_func.py
+++ com/automationpanda/example/calc_func.py
@@ -29,7 +29,7 @@

import math

-def logarithm(a, base=10):
+def logarithm(a, base=11):
    """
```

Retorna o logaritmo de 'a' na base especificada.
Levanta ValueError se 'a' <= 0 ou base <= 1.

Mutant 14

```
--- com/automationpanda/example/func.py
+++ com/automationpanda/example/func.py
@@ -34,7 +34,7 @@
     Retorna o logaritmo de 'a' na base especificada.
     Levanta ValueError se 'a' <= 0 ou base <= 1.
     """
-    if a <= 0:
+    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.")
```

Mutant 15

```
--- com/automationpanda/example/func.py
+++ com/automationpanda/example/func.py
@@ -34,7 +34,7 @@
     Retorna o logaritmo de 'a' na base especificada.
     Levanta ValueError se 'a' <= 0 ou base <= 1.
     """
-    if a <= 0:
+    if a <= 1:
         raise ValueError("O logaritmo só é definido para números positivos.")
     if base <= 1:
         raise ValueError("A base do logaritmo deve ser maior que 1.")
```

Mutant 16

```
--- com/automationpanda/example/func.py
+++ com/automationpanda/example/func.py
@@ -35,7 +35,7 @@
     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 18

```
--- com/automationpanda/example/func.py
+++ com/automationpanda/example/func.py
@@ -36,6 +36,6 @@
     """
     if a <= 0:
         raise ValueError("O logaritmo só é definido para números positivos.")
-    if base <= 1:
+    if base <= 2:
         raise ValueError("A base do logaritmo deve ser maior que 1.")
     return math.log(a, base)
```

Mutant 19

```
--- com/automationpanda/example/func.py
+++ com/automationpanda/example/func.py
@@ -37,5 +37,5 @@
     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)
```