```
from swarms.utils.try_except_wrapper import try_except_wrapper


def test_try_except_wrapper_with_no_exception():
    @try_except_wrapper
    def add(x, y):
        return x + y

    result = add(1, 2)
    assert (
        result == 3
    ), "The function should return the sum of the arguments"


def test_try_except_wrapper_with_exception():
    @try_except_wrapper
    def divide(x, y):
        return x / y

    result = divide(1, 0)
    assert (
        result is None
    ), "The function should return None when an exception is raised"


def test_try_except_wrapper_with_multiple_arguments():
    @try_except_wrapper
    def concatenate(*args):
        return "".join(args)

    result = concatenate("Hello", " ", "world")
    assert (
        result == "Hello world"
    ), "The function should concatenate the arguments"


def test_try_except_wrapper_with_keyword_arguments():
    @try_except_wrapper
    def greet(name="world"):
        return f"Hello, {name}"

    result = greet(name="Alice")
    assert (
        result == "Hello, Alice"
    ), "The function should use the keyword arguments"
```