



**WYŻSZA SZKOŁA
INFORMATYKI I ZARZĄDZANIA**
z siedzibą w Rzeszowie

Dokumentacja projektu

Przedmiot: **Metodyki zarządzania projektami
programistycznymi**

(Projekt z zakresu metodyki TDD)

Tytuł projektu:

Aplikacja monitorująca magazyn sklepowy

Prowadzący:
Dr Marek Jaszuk

Wykonawca: Gabriela Gurgul

Semestr, symbol kierunku i grupa:
UIZ-3/2020/IPO SP02

Rzeszów, 2022

1. Specyfikacja funkcjonalności.

Stworzona aplikacja ma symulować magazyn w sklepie obuwniczym. Jej użytkownik może dodawać nowe produkty oraz usuwać je w momencie wykupienia ich przez klienta. Dzięki aplikacji sklep może w łatwy sposób monitorować stan magazynu.

2. Specyfikacja przypadków testowych.

W celu stworzenia działającej w pełni aplikacji potrzebne są do pokrycia kolejne przypadki testowe:

- Dodawanie 5 par butów w cenie 10\$
- Dodanie do magazynu 0 par butów
- Dodanie 25 par butów do magazynu
- Usuwanie z magazynu butów, gdy ich ilość wynosi 0, 25 lub 5

3. Opis poszczególnych kroków wykonania programu w metodyce TDD z uwzględnieniem refaktoryzacji oraz pokrycia kodu testami.

Tworzenie aplikacji rozpoczęto od stworzenia obiektu Inventory oraz dodania do niego przykładowych obiektów – produktów w sklepie. Limit obiektów, które mogą być przechowywane w Inventory wynosi 100 (na potrzeby testów wartość tę zmieniono na 10).

```
def test_buy_and_sell_nikes_adidas():
    inventory = Inventory()
    assert inventory.limit == 100
    assert inventory.total_items == 0

    # Add new Nike sneakers
    inventory.add_new_stock('Nike Sneakers', 50.00, 10)
    assert inventory.total_items == 10

    # Add new Adidas sneakers
    inventory.add_new_stock('Adidas Sneakers', 70.00, 5)
    assert inventory.total_items == 15

    # Remove 2 sneakers to sell
    inventory.remove_stock('Nike Sneakers', 2)
    assert inventory.total_items == 13

    # Remove 1 shoeas to sell
    inventory.remove_stock('Adidas Sneakers', 1)
    assert inventory.total_items == 12

def test_default_inventory():
    """Test that the default limit is 100"""
    inventory = Inventory()
    assert inventory.limit == 100
    assert inventory.total_items == 0

class Inventory:
    def __init__(self, limit=100):
        self.limit = limit
        self.total_items = 0

def test_custom_inventory_limit():
    """Test that we can set a custom limit"""
    inventory = Inventory(limit=25)
    assert inventory.limit == 25
    assert inventory.total_items == 0
```

Test cases:

1. Dodawanie 5 par butów w cenie 10\$ do magazynu

```
def test_add_new_stock_success(no_stock_inventory):
    no_stock_inventory.add_new_stock('Test Shoes', 10.00, 5)
    assert no_stock_inventory.total_items == 5
    assert no_stock_inventory.stocks['Test Shoes']['price'] == 10.00
    assert no_stock_inventory.stocks['Test Shoes']['quantity'] == 5
```

Wynik testu:

```
===== FAILURES =====
test_add_new_stock_success
no_stock_inventory = <test.Inventory object at 0x7f98d4a95f70>
def test_add_new_stock_success(no_stock_inventory):
> no_stock_inventory.add_new_stock('Test Jacket', 10.00, 5)
E   AttributeError: 'Inventory' object has no attribute 'add_new_stock'
test.py:9: AttributeError
===== short test summary info =====
FAILED test.py::test_add_new_stock_success - AttributeError: 'Inventory' object has no attribute 'add_new_stock'
===== 1 failed in 0.28s =====
```

Aby przejść powyższy test należało dodać poniższy kod:

```
def add_new_stock(self, name, price, quantity):
    self.stocks[name] = {
        'price': price,
        'quantity': quantity
    }
    self.total_items += quantity
```

Wynik ponownego testu:

```
===== test session starts =====
platform darwin -- Python 3.9.0, pytest-7.1.2, pluggy-1.0.0
rootdir: /Users/gabrielagurgul/Documents/my/studia/projekt-python
collected 1 item

test.py . [100%]

===== 1 passed in 0.02s =====
```

Refaktoryzacja nie była potrzebna.

2. Dodanie do magazynu butów, których ilość = 0

```
@pytest.mark.parametrize('name,price,quantity,exception', [
    ('Test Shoes', 10.00, 0, InvalidQuantityException(
        'Cannot add a quantity of 0. Value must be at least 1 item'))
])

def test_add_new_stock_bad_input(name, price, quantity, exception):
    inventory = Inventory(10)
    try:
        inventory.add_new_stock(name, price, quantity)
    except InvalidQuantityException as inst:
        assert isinstance(inst, type(exception))
        assert inst.args == exception.args
    else:
        pytest.fail("Expected error but found none")
```

Wynik testu:

```
===== ERRORS =====
ERROR collecting test.py
test.py:13: in <module>
    class Inventory:
test.py:27: in Inventory
    ('Test Jacket', 10.00, 0, InvalidQuantityException(
E   NameError: name 'InvalidQuantityException' is not defined
===== short test summary info =====
ERROR test.py - NameError: name 'InvalidQuantityException' is not defined
!!!!!!!!!!!!!!!!!!!! Interrupted: 1 error during collection !!!!!!!!!!!!!!!!!!!!!
===== 1 error in 0.34s =====
```

Należy uzupełnić kod o poniższą klasę:

```
class InvalidQuantityException(Exception):
    pass
```

oraz zmienić metodę add_new_stock(...):

```
def add_new_stock(self, name, price, quantity):
    if quantity <= 0:
        raise InvalidQuantityException(
            'Cannot add a quantity of {}. Value must be at least 1'.format(quantity))
    self.stocks[name] = {
        'price': price,
        'quantity': quantity
    }
    self.total_items += quantity
```

Ponowne przeprowadzenie testu:

```
===== test session starts =====
platform darwin -- Python 3.9.0, pytest-7.1.2, pluggy-1.0.0
rootdir: /Users/gabrielagurgul/Documents/my/studia/projekt-python
collected 1 item

test.py . [100%]

===== 1 passed in 0.05s =====
```

Refaktoryzacja: zmieniono tekst wyświetlany na konsoli.

3. Dodaj 25 nowych par butów, limit magazynu wynosi 10.

```
@pytest.mark.parametrize('name,price,quantity,exception', [
    ('Test Shoes', 10.00, 0, InvalidQuantityException(
        'Cannot add a quantity of 0. Value must be at least 1 item')),
    ('Test Shoes', 10.00, 25, NoSpaceException(
        'Cannot add these 25 items. Only 10 items can be added'))
])
```

Wynik testu:

```
test.py:16: in <module>
    class Inventory:
test.py:35: in Inventory
    ('Test Jacket', 10.00, 25, NoSpaceException(
E   NameError: name 'NoSpaceException' is not defined

===== short test summary info =====
ERROR test.py - NameError: name 'NoSpaceException' is not defined
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Interrupted: 1 error during collection !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
===== 1 error in 0.40s =====
```

Należy zmienić wygląd metody `add_new_stock(...)`:

```
def add_new_stock(self, name, price, quantity):
    if quantity <= 0:
        raise InvalidQuantityException(
            'Cannot add a quantity of {}. All new stocks must have at least 1 item'.format(quantity))
    if self.total_items + quantity > self.limit:
        remaining_space = self.limit - self.total_items
        raise NoSpaceException(
            'Cannot add these {} items. Only {} more items can be stored'.format(quantity, remaining_space))
    self.stocks[name] = {
        'price': price,
        'quantity': quantity
    }
    self.total_items += quantity
```

oraz dodać nowy wyjątek:

```
class NoSpaceException(Exception):
    pass
```

Wynik testu:

```
===== test session starts =====
platform darwin -- Python 3.9.0, pytest-7.1.2, pluggy-1.0.0
rootdir: /Users/gabrielagurgul/Documents/my/studia/projekt-python
collected 1 item

test.py . [100%]

===== 1 passed in 0.02s =====
```

Refaktoryzacja:

Usunięcie niepotrzebnych komentarzy, wyrównanie kodu oraz przeniesiono wyjątki do pliku `inventory.py`, aby rozwiązanie było bardziej eleganckie i poprawne:

```
from inventory import Inventory, InvalidQuantityException, NoSpaceException
```

4. Usun 0 par butów Puma
Usun 25 par butów Puma
Usun 5 par butów Puma

Funkcja do usuwania rzeczy z magazynu:

```
def remove_stock(self, name, quantity):
    if quantity <= 0:
        raise InvalidQuantityException(
            'Cannot remove a quantity of {}. Must remove at least 1 item'.format(quantity))
    if name not in self.stocks:
        raise ItemNotFoundException(
            'Could not find {} in our stocks. Cannot remove non-existing stock'.format(name))
    if self.stocks[name]['quantity'] - quantity <= 0:
        raise InvalidQuantityException(
            'Cannot remove these {} items. Only {} items are in stock'.format(
                quantity, self.stocks[name]['quantity']))
    self.stocks[name]['quantity'] -= quantity
    self.total_items -= quantity
```

Test:

```
@pytest.mark.parametrize('name,quantity,exception,new_quantity,new_total', [
    ('Puma Test', 0,
     InvalidQuantityException(
         'Cannot remove a quantity of 0. Must remove at least 1 item'),
     0, 0),
    ('Not Here', 5,
     ItemNotFoundException(
         'Could not find Not Here in our stocks. Cannot remove non-existing stock'),
     0, 0),
    ('Puma Test', 25,
     InvalidQuantityException(
         'Cannot remove these 25 items. Only 8 items are in stock'),
     0, 0),
    ('Puma Test', 5, None, 3, 5)
```

Wynik testu:

```
ERROR collecting test.py
ImportError while importing test module '/Users/gabrielagurgul/Documents/my/studia/projekt-python/test.py'.
Hint: make sure your test modules/packages have valid Python names.
Traceback:
/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/importlib/__init__.py:127: in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
test.py:2: in <module>
    from inventory import Inventory, InvalidQuantityException, NoSpaceException, ItemNotFoundException
E   ImportError: cannot import name 'ItemNotFoundException' from 'inventory' (/Users/gabrielagurgul/Documents/my/studia/projekt-python/inventory.py)

===== short test summary info =====
ERROR test.py
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Interrupted: 1 error during collection !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
===== 1 error in 0.15s =====
```

Należało dodać kolejny wyjątek do pliku inventory.py

Końcowy test całego projektu:

```

gabriellagurgul@Gabrielas-MacBook-Pro projekt-python % pytest test.py
===== test session starts =====
platform darwin -- Python 3.9.0, pytest-7.1.2, pluggy-1.0.0
rootdir: /Users/gabriellagurgul/Documents/my/studia/projekt-python
collected 1 item

test.py . [100%]
===== 1 passed in 0.04s =====

gabriellagurgul@Gabrielas-MacBook-Pro projekt-python % pytest inventory.py
===== test session starts =====
platform darwin -- Python 3.9.0, pytest-7.1.2, pluggy-1.0.0
rootdir: /Users/gabriellagurgul/Documents/my/studia/projekt-python
collected 1 item

inventory.py . [100%]
===== 1 passed in 0.01s =====

```

4. Link do repozytorium z kodem projektu (kolejne iteracje implementacji zapisywać w repozytorium).

<https://github.com/gabriellagurgul/TDD>