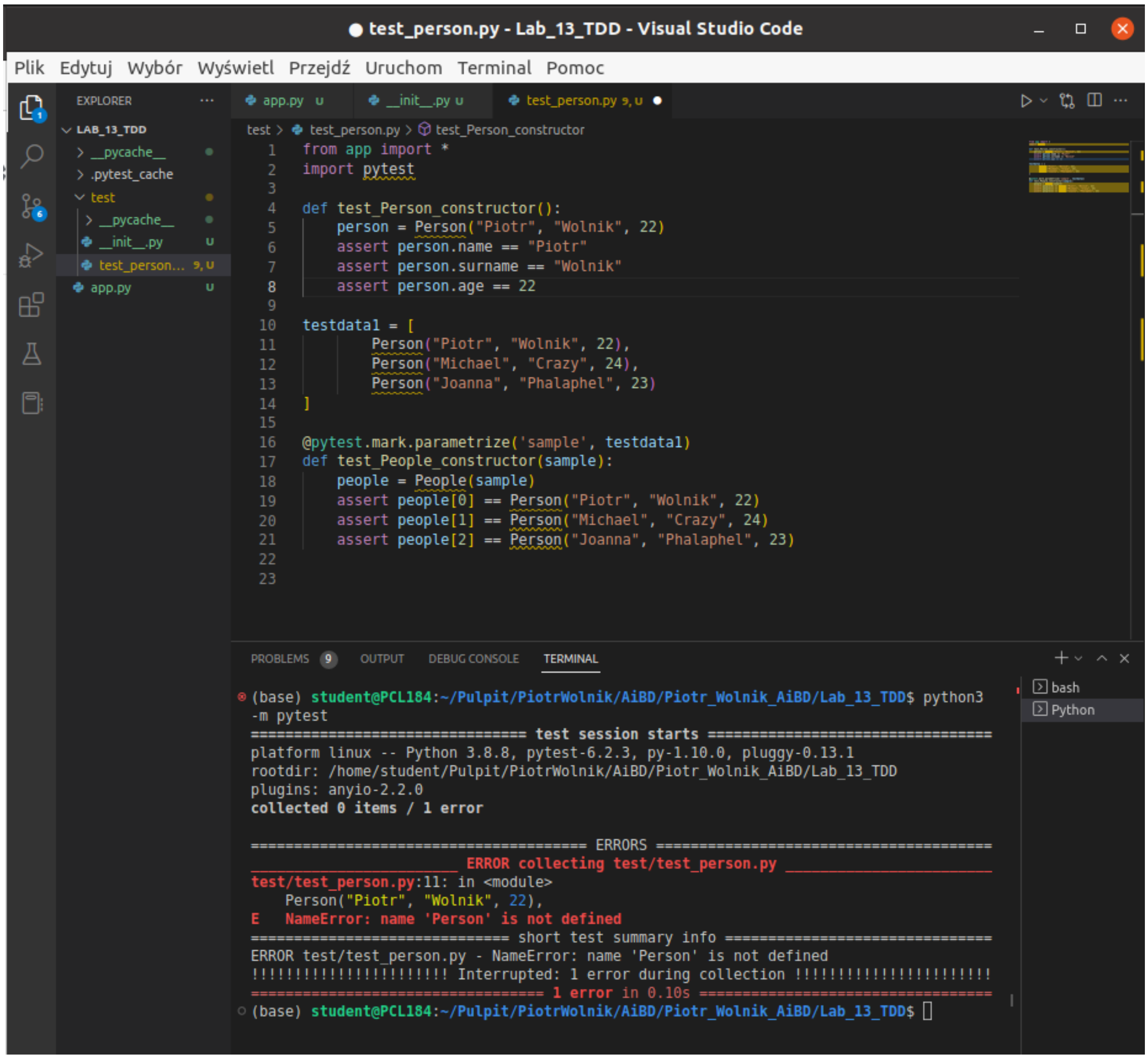
In my case I decided to use TDD on problem that was involving creating class Person and People. The first one, is supposed to represent a structure which attributes are: first name, second name and age. We have possibility of comparing objects of that class and printing in a special format it’s data. Second class is working like a container for Person class objects’. Options such as getting access to specific element and printing on the console are available.

Firstly, I started by creating simple tests that are described below (no implementation at that point was provided):

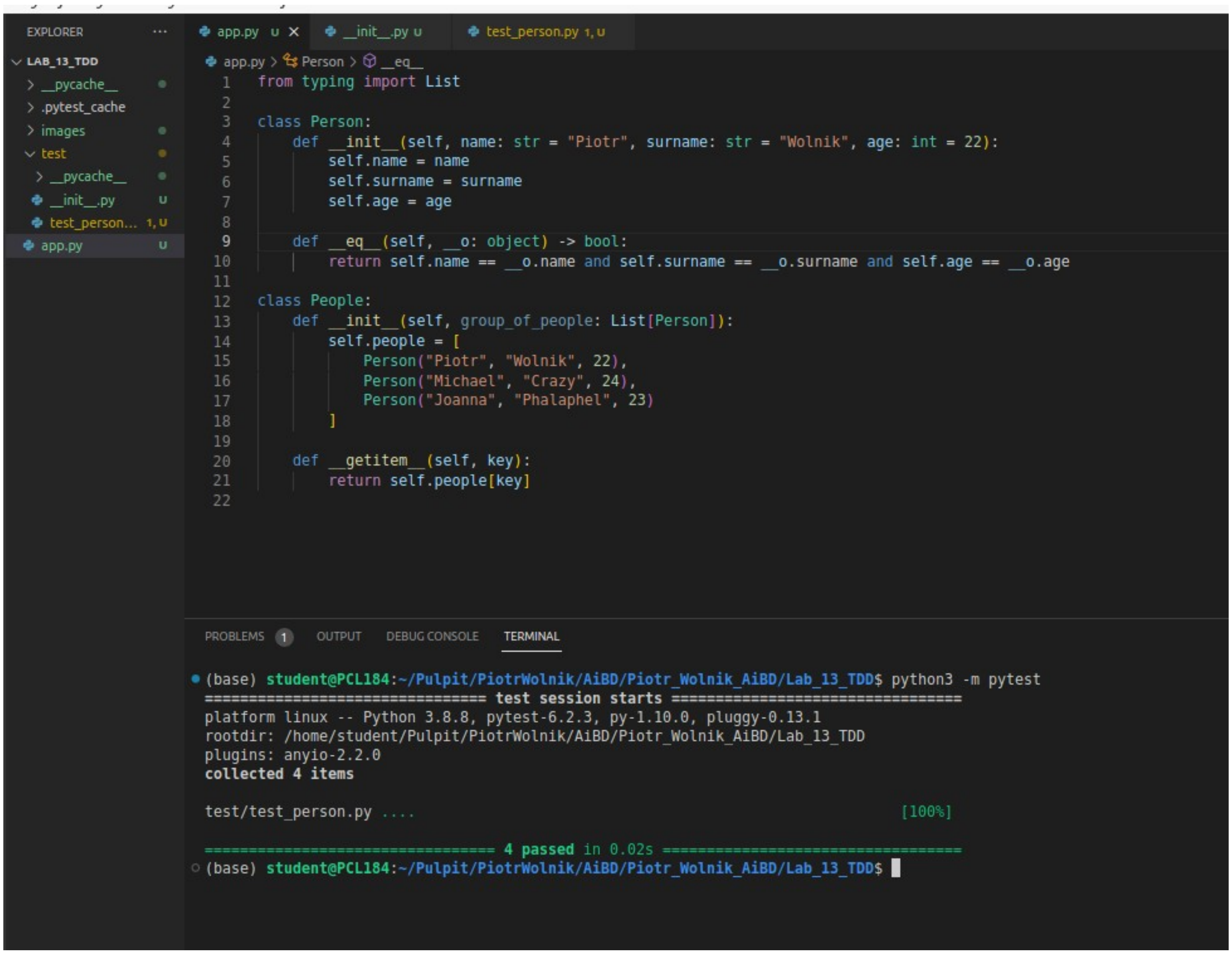


```python
from app import *
import pytest

def test_Person_constructor():
    person = Person("Piotr", "Wolnik", 22)
    assert person.name == "Piotr"
    assert person.surname == "Wolnik"
    assert person.age == 22

testdata1 = [
        Person("Piotr", "Wolnik", 22),
        Person("Michael", "Crazy", 24),
        Person("Joanna", "Phalaphel", 23)
]

@pytest.mark.parametrize('sample', testdata1)
def test_People_constructor(sample):
    people = People(sample)
    assert people[0] == Person("Piotr", "Wolnik", 22)
    assert people[1] == Person("Michael", "Crazy", 24)
    assert people[2] == Person("Joanna", "Phalaphel", 23)
```

```
(base) student@PCL184:~/Pulpit/PiotrWolnik/AiBD/Piotr_Wolnik_AiBD/Lab_13_TDD$ python3
-m pytest
================================ test session starts =================================
platform linux -- Python 3.8.8, pytest-6.2.3, py-1.10.0, pluggy-0.13.1
rootdir: /home/student/Pulpit/PiotrWolnik/AiBD/Piotr_Wolnik_AiBD/Lab_13_TDD
plugins: anyio-2.2.0
collected 0 items / 1 error

======================================= ERRORS =======================================
________________________ ERROR collecting test/test_person.py ________________________
test/test_person.py:11: in <module>
    Person("Piotr", "Wolnik", 22),
E   NameError: name 'Person' is not defined
============================== short test summary info ===============================
ERROR test/test_person.py - NameError: name 'Person' is not defined
!!!!!!!!!!!!!!!!!!!!!!! Interrupted: 1 error during collection !!!!!!!!!!!!!!!!!!!!!!!
=================================== 1 error in 0.10s ===================================
(base) student@PCL184:~/Pulpit/PiotrWolnik/AiBD/Piotr_Wolnik_AiBD/Lab_13_TDD$ 
```

As we can see we test constructors of both class to check if objects are created in a proper way. The second test involves using undefined index operator.

Then, we go straight to phase GREEN which stands for writing minimal implementation so that the tes ts would pass:

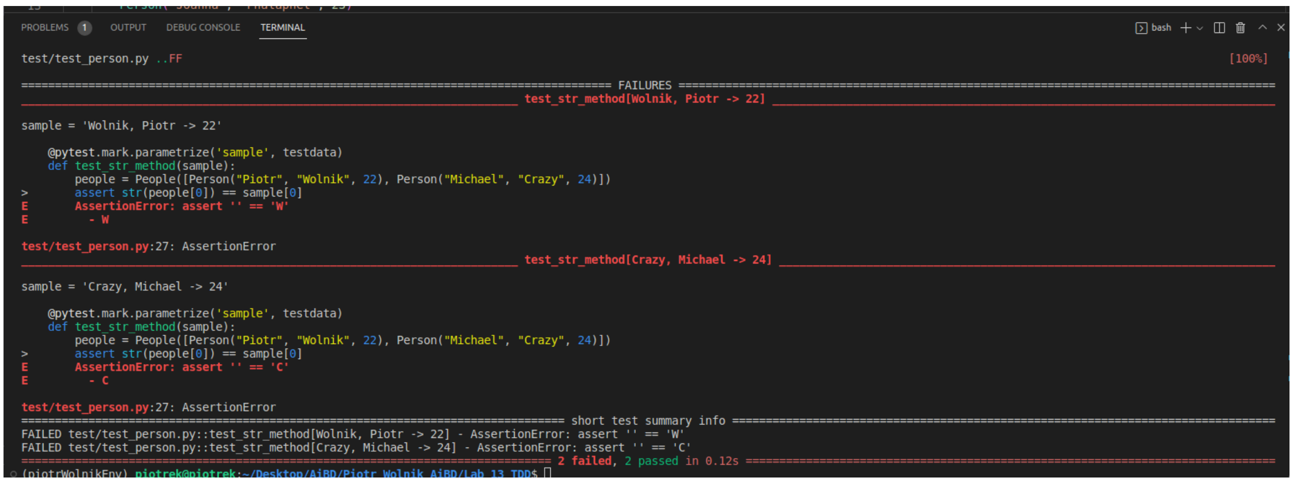


```python
from typing import List

class Person:
    def __init__(self, name: str = "Piotr", surname: str = "Wolnik", age: int = 22):
        self.name = name
        self.surname = surname
        self.age = age

    def __eq__(self, __o: object) -> bool:
        return self.name == __o.name and self.surname == __o.surname and self.age == __o.age

class People:
    def __init__(self, group_of_people: List[Person]):
        self.people = [
            Person("Piotr", "Wolnik", 22),
            Person("Michael", "Crazy", 24),
            Person("Joanna", "Phalaphel", 23)
        ]

    def __getitem__(self, key):
        return self.people[key]

```

```
(base) student@PCL184:~/Pulpit/PiotrWolnik/AiBD/Piotr_Wolnik_AiBD/Lab_13_TDD$ python3 -m pytest
=============================== test session starts ================================
platform linux -- Python 3.8.8, pytest-6.2.3, py-1.10.0, pluggy-0.13.1
rootdir: /home/student/Pulpit/PiotrWolnik/AiBD/Piotr_Wolnik_AiBD/Lab_13_TDD
plugins: anyio-2.2.0
collected 4 items

test/test_person.py ....                                                    [100%]

================================= 4 passed in 0.02s =================================
(base) student@PCL184:~/Pulpit/PiotrWolnik/AiBD/Piotr_Wolnik_AiBD/Lab_13_TDD$ 
```

All tests pass.

Next, we provide some new test that is responsible for checking if __str__ function of Person class is working correctly. Of course at this point we provide no implementation of that function either (we just define it):

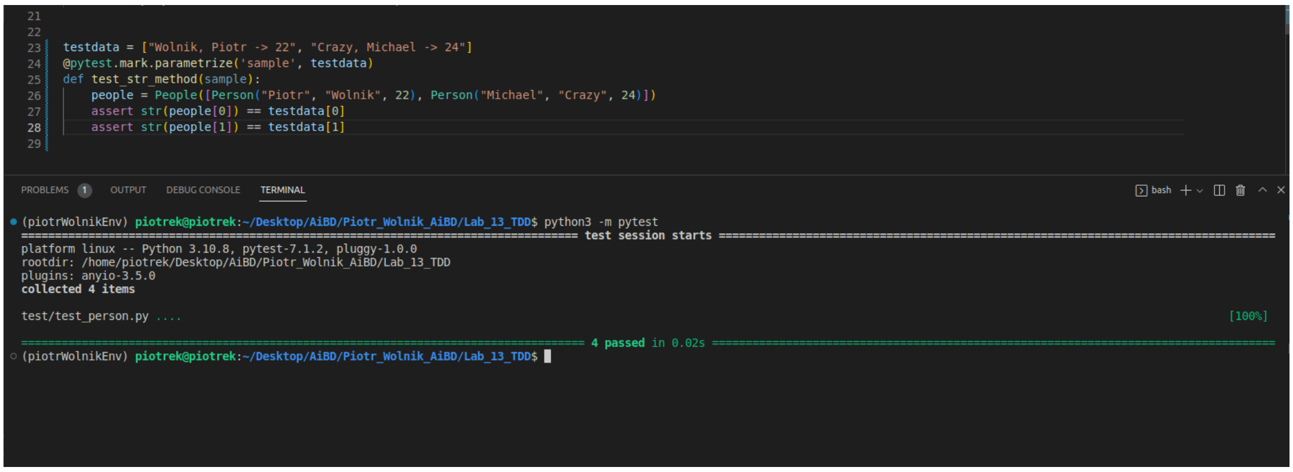


```
test/test_person.py ..FF                                                                                   [100%]

===================================================================================== FAILURES =====================================================================================
___________________________________________________________________________ test_str_method[Wolnik, Piotr -> 22] ___________________________________________________________________________

sample = 'Wolnik, Piotr -> 22'

    @pytest.mark.parametrize('sample', testdata)
    def test_str_method(sample):
        people = People([Person("Piotr", "Wolnik", 22), Person("Michael", "Crazy", 24)])
>       assert str(people[0]) == sample[0]
E       AssertionError: assert '' == 'W'
E         - W

test/test_person.py:27: AssertionError
___________________________________________________________________________ test_str_method[Crazy, Michael -> 24] ___________________________________________________________________________

sample = 'Crazy, Michael -> 24'

    @pytest.mark.parametrize('sample', testdata)
    def test_str_method(sample):
        people = People([Person("Piotr", "Wolnik", 22), Person("Michael", "Crazy", 24)])
>       assert str(people[0]) == sample[0]
E       AssertionError: assert '' == 'C'
E         - C

test/test_person.py:27: AssertionError
============================================================================ short test summary info ============================================================================
FAILED test/test_person.py::test_str_method[Wolnik, Piotr -> 22] - AssertionError: assert '' == 'W'
FAILED test/test_person.py::test_str_method[Crazy, Michael -> 24] - AssertionError: assert '' == 'C'
========================================================================= 2 failed, 2 passed in 0.12s =========================================================================
(piotrWolnikEnv) piotrek@piotrek:~/Desktop/AiBD/Piotr_Wolnik_AiBD/Lab_13_TDD$ 
```

As we can see, test fails.

The overall cycle of TDD development is met.

Nevertheless, we want to have no tests that failed, so we provide implementation for __str__ function of Person class with the intention that all assertions are correct:



```python


testdata = ["Wolnik, Piotr -> 22", "Crazy, Michael -> 24"]
@pytest.mark.parametrize('sample', testdata)
def test_str_method(sample):
    people = People([Person("Piotr", "Wolnik", 22), Person("Michael", "Crazy", 24)])
    assert str(people[0]) == testdata[0]
    assert str(people[1]) == testdata[1]

```

```
(piotrWolnikEnv) piotrek@piotrek:~/Desktop/AiBD/Piotr_Wolnik_AiBD/Lab_13_TDD$ python3 -m pytest
=========================================================================== test session starts ===========================================================================
platform linux -- Python 3.10.8, pytest-7.1.2, pluggy-1.0.0
rootdir: /home/piotrek/Desktop/AiBD/Piotr_Wolnik_AiBD/Lab_13_TDD
plugins: anyio-3.5.0
collected 4 items

test/test_person.py ....                                                                                                                                            [100%]

============================================================================ 4 passed in 0.02s ============================================================================
(piotrWolnikEnv) piotrek@piotrek:~/Desktop/AiBD/Piotr_Wolnik_AiBD/Lab_13_TDD$ 
```