**Лабораторна робота №2**

**ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Мета: розглянути принципи модульного тестування програмного забезпечення; навчитися створювати модульні тести з використанням unittest.

Хід роботи:

**Варіант 11.** Реалізувати сервіс/програму ведення запису і розподілу дітей у початкові класи школи

З файлу **my\_program.py**

# checks if input is right. Example of right input: '6: Ben Konrad'

def check\_format(arg):

    if arg[0]=='6' or arg[0]=='7' or arg[0]=='8' and len(arg)>= 4:

        return True

    return False

# adds ':' character if it isn't there

def two\_dot\_formating(arg):

    if arg[1]!=':':

        return str(arg[:1] + ':' + arg[1:])

    return arg

# adds ' ' character if needed

def space\_formating(arg):

    if arg[2]!=' ':

        return str(arg[:2] + ' ' + arg[2:])

    return arg

# distributes children amongst classes depending on their age

def distrib\_by\_age(arg):

    if int(arg[0]) == 6:

        return str(arg + ' -> 1st grade')

    elif int(arg[0]) == 7:

        return str(arg + ' -> 2nd grade')

    elif int(arg[0]) == 8:

        return str(arg + ' -> 3rd grade')

    else:

        return str(arg + ' -> Incorrect age')

def exception(arg):

    if int(arg[0]) == 5:

        raise Exception("Age exception | 5 years -> 1st grade")

    return arg

**З файлу my\_test.py**

import unittest

import my\_program as prog

class TestMyProgram(unittest.TestCase):

    # basic setup

    def setUp(self):

        self.prog = prog

    # test for checking format

    def test\_check\_format(self):

        self.assertTrue(prog.check\_format('6Ben Schaefer'))     # correct

        self.assertTrue(prog.check\_format('7Findlay Shaffer'))  # correct

        self.assertTrue(prog.check\_format('8Kara Moss'))        # correct

        self.assertFalse(prog.check\_format('Pen'))              # no age + incorrect length

        self.assertFalse(prog.check\_format('Emre Walls'))       # no age

        self.assertFalse(prog.check\_format('5Asa Rojas'))       # incorrect age

        self.assertFalse(prog.check\_format('9Casey Long'))      # incorrect age

        self.assertFalse(prog.check\_format('9E'))               # incorrect length

    # test for adding ':' character if needed

    def test\_two\_dot\_formating(self):

        self.assertEqual(prog.two\_dot\_formating('6Anna Rou'), '6:Anna Rou')          # changed

        self.assertEqual(prog.two\_dot\_formating('6:Reg Realmer'), '6:Reg Realmer')   # unchanged

    # test for adding ' ' character if needed

    def test\_space\_formatting(self):

        self.assertEqual(prog.space\_formating('8:Jake Snow'), '8: Jake Snow')      # changed

        self.assertEqual(prog.space\_formating('7: Lyla Dylla'), '7: Lyla Dylla')   # unchanged

    # test group distribution

    def test\_distrib\_by\_age(self):

        self.assertEqual(prog.distrib\_by\_age('6: Andreas Pittman'), '6: Andreas Pittman -> 1st grade')

        self.assertEqual(prog.distrib\_by\_age('7: Blake Bowman'), '7: Blake Bowman -> 2nd grade')

        self.assertEqual(prog.distrib\_by\_age('8: Ruth Roy'), '8: Ruth Roy -> 3rd grade')

        self.assertEqual(prog.distrib\_by\_age('5: Katie Rivera'), '5: Katie Rivera -> Incorrect age')

    def test\_exception(self):

        with self.assertRaises(Exception):

            prog.exception('5: Dyllan Bradly')   # returns true exception occurs

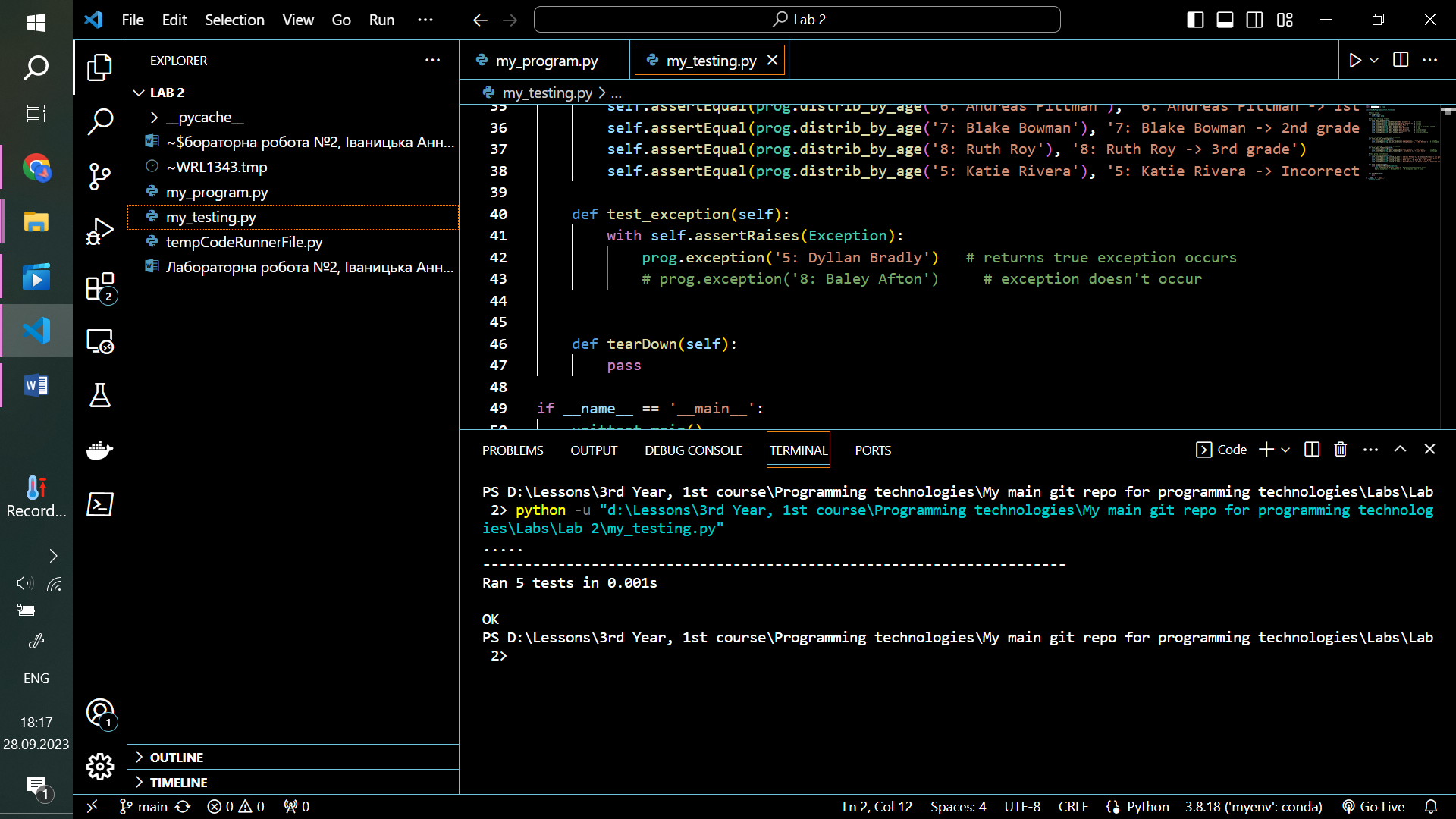
            # prog.exception('8: Baley Afton')     # exception doesn't occur

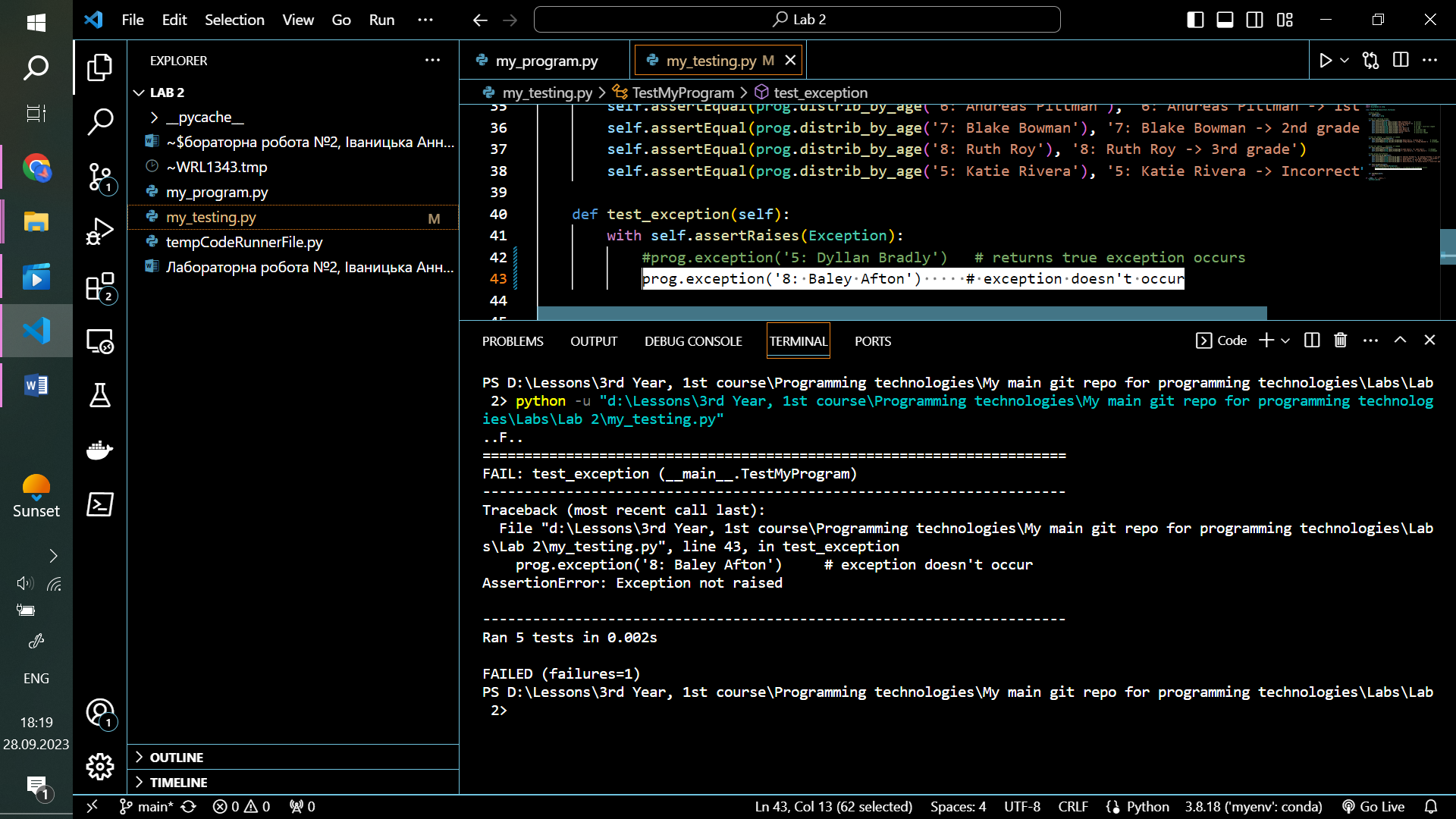
    def tearDown(self):

        pass

if \_\_name\_\_ == '\_\_main\_\_':

    unittest.main()

Вивід виконання тестування програми

(вивід, коли exception не виникає)

Висновок: У ході виконання цієї лабораторної роботи я навчилася писати unit тести для програм.