Introdução à testes automatizados em PyCharm

Tarefa 1: Criar projeto

https://www.jetbrains.com/help/pycharm/creating-and-running-your-first-python-project.html#edit-file

Tarefa 2
Preparar código

```python
class Car:

    def __init__(self, acc=5, brake=3):
        self._acc = acc
        self._brake = brake
        self.speed = 0.
        self.odometer = 0
        self.time = 0

    def say_state(self):
        print("I'm going {} kph!".format(self.speed))

    def accelerate(self):
        self.speed += self._acc

    def brake(self):
        self.speed -= self._brake

    def step(self):
        self.odometer += self.speed
        self.time += 1

    def average_speed(self):
        if self.time != 0:
            return self.odometer / self.time
        else:
            pass
```

```python
class TestDrive:
    def __init__(self, car):
        self.car = car

    def start(self):
        while True:
            print(f'{"=" * 40} t{self.car.time} {"=" * 40}')
            action = input("What should I do? [A]ccelerate, [B]rake, "
                           "show [O]dometer, show average [S]peed or sto[P]?").upper()
            if action not in "ABOSP" or len(action) != 1:
                print("I don't know how to do that")
                continue
            if action == 'A':
                self.car.accelerate()
            elif action == 'B':
                self.car.brake()
            elif action == 'O':
                print("The car has driven {} kilometers".format(my_car.odometer))
            elif action == 'S':
                print("The car's average speed was {} kph".format(self.car.average_speed()))
            elif action == 'P':
                print("The test finished!")
                break
            self.car.step()
            self.car.say_state()
            print('\n')
```

```python
if __name__ == '__main__':

    my_car = Car(acc=x, brake=y)
    test = TestDrive(my_car)
    test.start()
```

Tarefa 3

a) Rodar
b) Debugar

Teste se:

I. o carro está acelerando

II. o carro está freiando

III. o tempo está correndo

IV. o odômetro está funcionando

- Step Into (F7)
- Step Over (F8)
- Step Out (Shift+ F8)

[menu de contexto]

- Run to cursor
- Jump to cursor

- Breakpoint
- Breakpoint condicional

Qual o valor do odômetro após acelerar 3x, frear 2x, e acelerar 1x?

Tarefa 4

testes

```python
import unittest

from my_classes.Car import Car


class TestCar(unittest.TestCase):
    def setUp(self):
        self.car = Car()


class TestInit(TestCar):
    def test_initial_speed(self):
        self.assertEqual(self.car.speed, 0)

    def test_initial_odometer(self):
        self.assertEqual(self.car.odometer, 0)

    def test_initial_time(self):
        self.assertEqual(self.car.time, 0)


class TestAccelerate(TestCar):
    def test_accelerate_from_zero(self):
        self.car.accelerate()
        self.assertEqual(self.car.speed, X)

    def test_multiple_accelerates(self):
        for _ in range(3):
            self.car.accelerate()
        self.assertEqual(self.car.speed, X)
```

```python
class TestBrake(TestCar):
    def test_brake_once(self):
        self.car.accelerate()
        self.car.brake()
        self.assertEqual(self.car.speed, 0)

    def test_multiple_brakes(self):
        for _ in range(5):
            self.car.accelerate()
        for _ in range(3):
            self.car.brake()
        self.assertEqual(self.car.speed, X)

    def test_should_not_allow_negative_speed(self):
        self.car.brake()
        self.assertEqual(self.car.speed, 0)

    def test_multiple_brakes_at_zero(self):
        for _ in range(3):
            self.car.brake()
        self.assertEqual(self.car.speed, X)
```

```python
if __name__ == '__main__':
    unittest.main()
```