```python
from machine import Pin, ADC
import time

class default_moisture_sensor:
    PIN = 27

class MoistureSensor:
    @classmethod
    def get_default_moisture_sensor(cls):
        return MoistureSensor(default_moisture_sensor.PIN)

    def __init__(self, moisturePin:int = default_moisture_sensor.PIN):
        """
        Implements for a moisture sensor using the built in 12-bit ADC.
        Reads from analog in and converts to a int from 0 (white) to 100 (black)

        :param leftPin: The pin the left moisture sensor is connected to
        :type leftPin: int
        :param rightPin: The pin the right moisture sensor is connected to
        :type rightPin: int
        """
        self._sensor = ADC(Pin(moisturePin))

        self.MAX_ADC_VALUE: int = 65536

    def _get_value(self, sensor: ADC) -> float:
        return int((sensor.read_u16() / self.MAX_ADC_VALUE) * 100)

    def read(self) -> int:
        """
        Gets the the reflectance of the left reflectance sensor
        : return: The reflectance ranging from 0 (white) to 1 (black)
        : rtype: int
        """
        return self._get_value(self._sensor)

if __name__ == "__main__":
    sensor = MoistureSensor.get_default_moisture_sensor()
    try:
        while True:
            reading = sensor.read()
            print(f"Moisture reading: {reading}%")
            time.sleep(0.5)
    except KeyboardInterrupt:
        print("Stopped by user.")

```