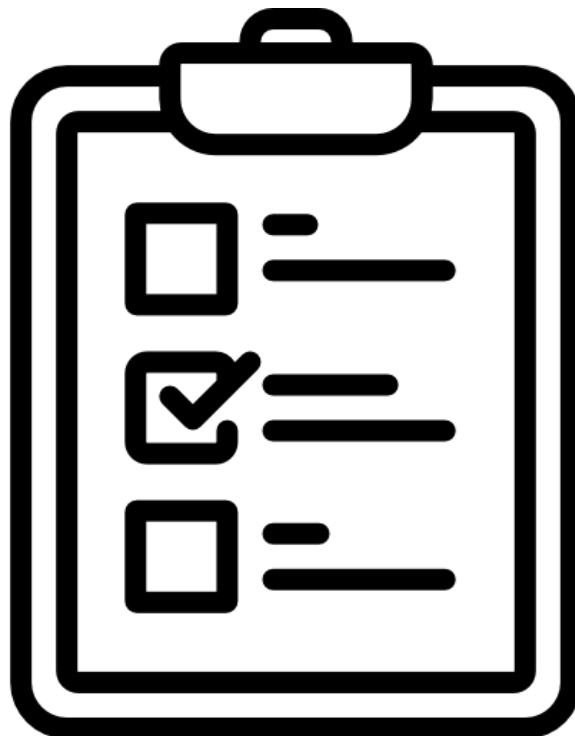




TEST RAPPORT

ATP – Opdracht 4



14 APRIL 2024

Naam: Berke Özmuk
Studentennummer: 1762463
Docent: Nick Goris
Vak: ATP

Inhoudsopgave

Test rapport regelsysteem.....	2
Unit tests	2
Doel	2
Motivatie	2
Resultaten	2
Integratietest.....	3
Doel	3
Motivatie	3
Resultaten	3
Systeemtest	4
Doel	4
Motivatie	4
Resultaat.....	4

Test rapport regelsysteem

Dit document beschrijft hoe de tests zijn uitgevoerd en welke impact deze hebben op de kwaliteit van het systeem. Alle tests worden uitgevoerd met het behulp van de python library "unittest".

Unit tests

Er zijn in totaal drie unittest gedaan:

```
class unit_test(unittest.TestCase):
    @calc_runtime
    def test_fahrenheit_to_celsius1(self):
        result = round(sensor.fahrenheitToCelsius(c_float(70.0)),1)
        self.assertEqual(result, 21.1, "Het resultaat moet 21.1 zijn!")

    @calc_runtime
    def test_fahrenheit_to_celsius2(self):
        result = round(sensor.fahrenheitToCelsius(c_float(75.0)),1)
        self.assertEqual(result, 23.9, "Het resultaat moet 23.9 zijn!")

    @calc_runtime
    def test_fahrenheit_to_celsius3(self):
        result = round(sensor.fahrenheitToCelsius(c_float(80.0)),1)
        self.assertEqual(result, 26.7, "Het resultaat moet 26.7 zijn!")
```

Er wordt vergeleken of de resultaat van de test overeenkomt met de juiste uitvoer.

Doel

Het doel van deze unit tests is om vast te stellen dat de conversie van Fahrenheit naar Celsius goed berekent wordt.

Motivatie

Deze test wordt gedaan, omdat het regelsysteem van het verwarmen van de kamer afhankelijk is van de temperatuur in graden. Zonder deze test uit te voeren kan er niet worden vastgesteld of de conversie goed gaat, waardoor dit een groot invloed kan hebben op het regelsysteem en op de functionele geschiktheid van het regelsysteem. Dit kan er namelijk voor zorgen dat het regelsysteem gaat verwarmen wanneer dit niet moet gebeuren of juist wel gaat verwarmen wanneer dit niet moet gebeuren.

Deze unit tests zijn ook van belang voor vervolg implementaties, omdat de gebruiker weet dat de temperatuur waarmee wordt gewerkt juist geconverteerd zijn.

Resultaten

```
berke@Schoool:~/ATP_BerkeOzmuk/Eindopdracht$ python3 -m unittest unit_test.py
test_fahrenheit_to_celsius1 runtime: 0:00:00.000019 seconds
.test_fahrenheit_to_celsius2 runtime: 0:00:00.000017 seconds
.test_fahrenheit_to_celsius3 runtime: 0:00:00.000006 seconds
.
-----
Ran 3 tests in 0.000s

OK
```

Integratietest

Er is één integratie test gedaan:

```
class integration_test(unittest.TestCase):
    @calc_runtime
    def test_set_kachel_on(self):
        temperature =
round(sensor.fahrenheitToCelsius(c_float(67.1)),1) #19.5 C
        round(sensor.setKachel(c_float(temperature)),1)
        result = sensor.getKachelStatus()
        self.assertEqual(result, True, "Het resultaat moet True zijn!")
```

Er wordt vergeleken of de status van de Kachel “True” is. True wilt zeggen dat de kachel is aangezet. Er wordt hardcoded 19.5°C meegegeven (op 19.5°C moet de kachel gaan verwarmen), zodat de kachel gaat verwarmen.

Doel

Het doel van deze integratietest is om vast te stellen dat de kachel gaat verwarmen als de temperatuur gelijk of onder 19.5°C valt.

Motivatie

Deze test wordt gedaan, zodat er vastgesteld kan worden dat de kachel op de juiste temperatuur gaat beginnen met verwarmen. Als deze test niet wordt uitgevoerd kan er niet worden vastgesteld of de kachel op de juiste temperaturen gaat verwarmen. Dit heeft invloed op de functionele geschiktheid van het regelsysteem. Met deze test waarborg je dat de kachel op de juiste temperaturen gaat beginnen met verwarmen.

Deze integratietest is ook van belang voor vervolg implementaties als de gebruiker gebruik wilt maken van de kachel, omdat de gebruiker aan de hand van deze tests kan zien dat de kachel op de correcte wijze werkt.

Resultaten

```
berke@Schoool:~/ATP_BerkeOzmuk/Eindopdracht$ python3 -m unittest integration_test.py
19.5kachel aan
test_set_kachel_on runtime: 0:00:00.000057 seconds
.
-----
Ran 1 test in 0.000s

OK
```

Systeemtest

Er is één systeemtest gedaan:

```
class system_test(unittest.TestCase):
    @calc_runtime
    def test_set_kachel_on_off(self):
        temperature =
round(sensor.fahrenheitToCelsius(c_float(67.1)),1) #19.5 C
        result = round(sensor.setKachel(c_float(temperature)),1)
        while sensor.getKachelStatus():
            result = round(sensor.setKachel(c_float(result)),1)
        self.assertEqual(result, 20.0, "Het resultaat moet 20.0 zijn!")
```

Er wordt vergeleken of de kachel verwarmt en stopt met verwarmen als de temperatuur 20°C graden heeft bereikt. Net als de integratietest wordt er hardcoded 19.5°C meegegeven.

Doel

Het doel van deze systeemtest is om vast te stellen dat de gehele werking van het systeem correct werkt. Er wordt getest of de koppeling tussen de temperatuur sensor en kachel goed werkt.

Motivatie

Deze test wordt gedaan, zodat er vastgesteld kan worden dat het regelsysteem tussen de koppeling van de temperatuur sensor en kachel correct werkt. Als dit niet wordt gedaan kan de betrouwbaarheid van het regelsysteem niet gewaarborgd worden. Deze test is van belang, omdat dit aantoont dat de gebruiker kan vertrouwen dat het regelsysteem op de juiste manier werkt en zijn taken realiseert.

Deze systeemtest is ook van belang voor vervolg implementaties, omdat de gebruiker kan zien dat het regelsysteem werkt. Op basis van deze test kan de gebruiker het regelsysteem uitbreiden zonder zich zorgen te hoeven maken of het regelsysteem wel op de juiste manier werkt.

Resultaat

```
berke@Schoool:~/ATP_BerkeOzmuk/Eindopdracht$ python3 -m unittest system_test.py
kachel aan
kachel uit
test_set_kachel_on_off runtime: 0:00:00.000222 seconds
.
-----
Ran 1 test in 0.000s

OK
```