Source code:

Hàm cần kiểm thử:

```python
    def get_net_cost(self):
        gross_cost = self.rate_per_min * self.duration
        if (self.validate_time()):
            # Discount if the call is in a certain given time of
the day
            if (self.start_time >= self.lower_bound and
self.start_time <= self.upper_bound):
                net_cost = gross_cost
            else:
                net_cost = gross_cost * self.discount_1

            # Any call longer than 60 minutes receives a 15%
discount on its cost
            if (self.duration >= 60):
                net_cost -= net_cost * self.discount_2

            # All calls are subject to a 4% Federal tax.
            tax = net_cost * self.tax
            net_cost += tax

            return round(net_cost, 2)
        else:
            return "Invalid input"
```

Các ca kiểm thử đảm bảo độ phủ cấp 2:

```python
import unittest
from cost_calc import Call
class TestCalcBoundaryValue(unittest.TestCase):
    def test_cost_1(self):
        self.call = Call(25, 31, 100)
        self.assertEqual(self.call.net_cost, 'Invalid input')
    def test_cost_2(self):
        self.call = Call(15, 31, 100)
        self.assertEqual(self.call.net_cost, 30.94)
    def test_cost_3(self):
        self.call = Call(22, 31, 10)
        self.assertEqual(self.call.net_cost, 1.82)
if __name__ == '__main__':
    unittest.main()
```

Control Flow Graph

```
      ①                          ②                                            No
entry → gross_cost =          start_time ─────────────────────────────────────────┐
        self.rate_per_min *    valid?                                             │
        self.duration                                                            ⑨
                                  │ Yes                                    return "Invalid input"
                                  ▼
                              ③                                                   │
        Yes             start_time in              No                            │
    ┌───────────────     normal  ───────────────┐                              │
    │                    range?                 │                               │
    ▼                                           ▼                               │
   ④                                          ⑤                                │
net_cost =                              net_cost =                              │
gross_cost                             gross_cost *                            │
                                       self.discount_1                         │
    │                                       │                                   │
    │                   ⑥                  │                                   │
    │         Yes   duration >= 60?         │                                   │
    │     ┌──────────                       │                                   │
    │     │            │ No                 │                                   │
   ⑦    │            │                     │                                   │
net_cost -=  │         │                     │                                   │
net_cost *   │         │                     │                                   │
self.discount_2        │        ⑧           │                                   │
                       ▼                     │                                   ▼
              net_cost += tax  ──────────────────────────────────────────────→ exit
              return
              round(net_cost, 2)
```
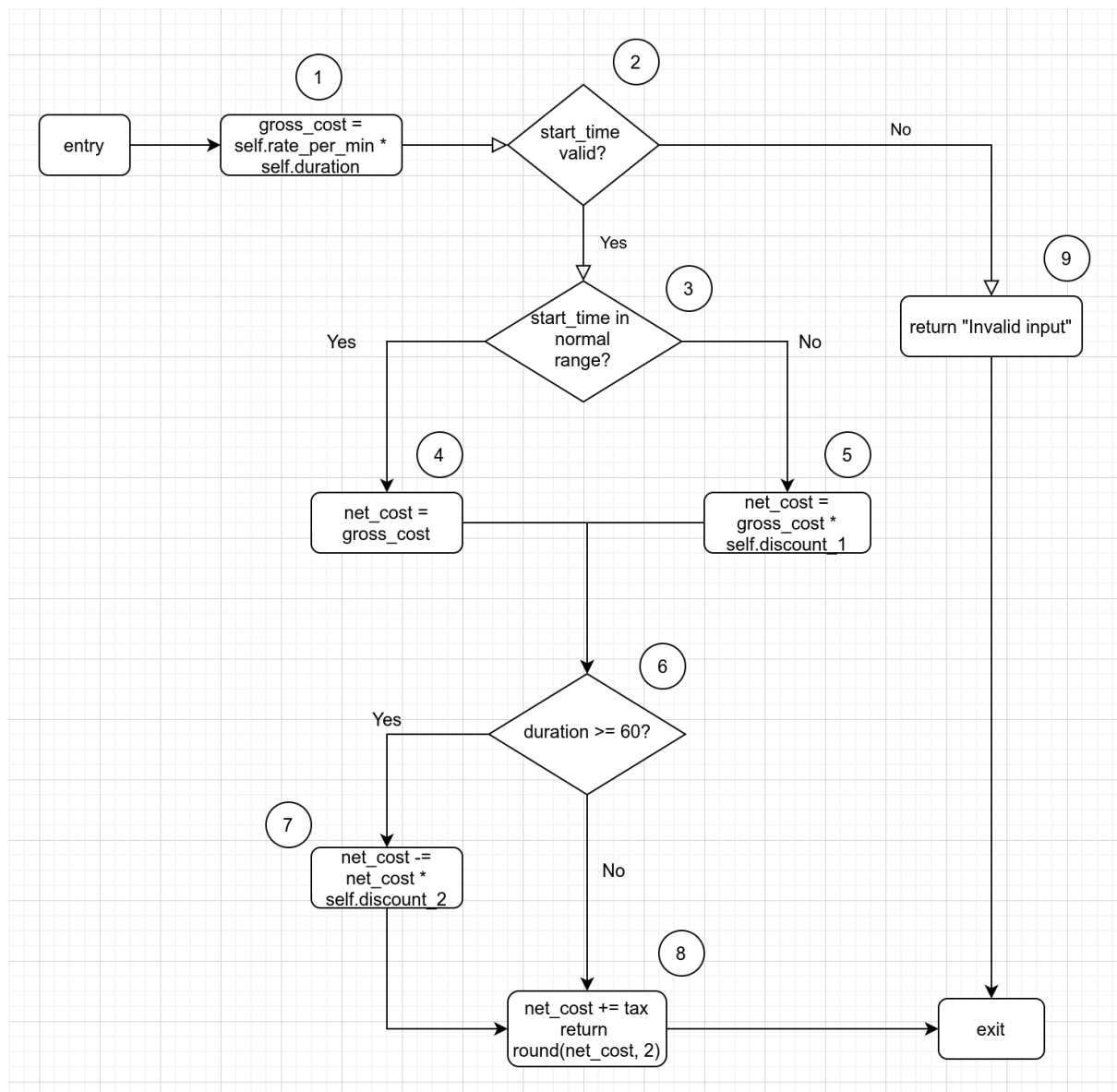
Test Paths, Test Cases (độ phủ cấp 2):

#Test	Test Path	Test Case (hh, mm, duration)	Expected	Actual	Status
1	1, 2(F), 9	(25, 31, 100)	"Invalid"	"Invalid"	PASSED
2	1, 2(T), 3(T), 4, 6(T), 7, 8	(15, 31, 100)	30.94	30.94	PASSED
3	1, 2(T), 3(F), 5, 6(F), 8	(22, 31, 10)	1.82	1.82	PASSED