

파이썬을 이용한 커피머신 동작 알고리즘

1) 고려 사항

자판기에 들어갈 값들

- * 보유 금액
- * 보유된 커피 수
- * 커피 가격

사용자가 사용할 값들

- * 입력 금액
- * 원하는 커피 수

내부에서 일어날 값들

- * 총 결제 금액
- * 줘야 할 거스름돈

main 함수는 최대한 간결하게

고려 할 경우의 수

돈을 충분히 넣은 경우

돈을 반정도 넣은 경우

돈을 불충분하게 넣은 경우

자판기 내의 돈이 부족한 경우

자판기 내의 남은 커피수가 부족한 경우

vi Coffee.py

```
class Coffee:
    total_amount = 10
    total_amount_price = 5000
    coffee_price = 300
    put_price = 0      # 결제 금액
    req_coffee_nums    = 0 # 원하는 커피 수
    remaining_price = 0 # 거스름돈
    #endpay = 0 # 총 결제 금액

# 사용자의 커피 돈, 입력
def request(self):
    self.put_price = int(input('돈을 넣어 주세요: '))
    self.req_coffee_nums = int(input('몇개를 사시겠습니까?: '))

# 커피 뽑는 동작
#돈이 충족하게 넣어졌을 경
def get(self, put_price, req_coffee_nums):
    import math
    # 총 결제 금액
    self.endpay = self.coffee_price * self.req_coffee_nums

    # 충분히 넣은 경우
    if self.put_price >= self.endpay:
        # 거스름돈이 없는 경우
        self.remaining_price = self.put_price - self.endpay
        if self.remaining_price == 0:
            print("커피를 %d개 줍니다" % self.req_coffee_nums)
            print("거스름돈 없음")
            self.total_amount -= self.req_coffee_nums
        # 거스름돈이 있는 경우
        else:
            print("커피를 %d개 줍니다" % self.req_coffee_nums)
            print("거스름돈은 %d 입니다" % self.remaining_price)
            self.total_amount -= self.req_coffee_nums
            self.total_amount_price -= self.endpay

    elif self.coffee_price <= self.put_price < self.endpay:
        n = math.trunc(self.put_price / self.coffee_price)
        self.remaining_price = put_price - (self.coffee_price * n)
```

```

        print("커피를 %d개 줍니다" % n)
        print("거스름돈은 %d 입니다" % self.remaining_price)
        self.total_amount -= n
        self.total_amount_price -= (self.coffee_price * n)

    else:
        print("[FAIL]돈이 부족하거나 갯수가 이상합니다.")
        print("거스름돈은 %d 입니다" % self.put_price)
        return False

# 자판기에 남아있는 돈과 커피
def info(self):
    print("*****")
    print('남은 커피 수: %d ' % self.total_amount)
    print('남은 금액: %d ' % self.total_amount_price)
    print("*****")

# 자판기에 남아있는 커피 갯수 점검
def check_amount(self):
    if self.total_amount <= 0 or self.total_amount_price < 0:
        if self.total_amount <= 0:
            print("*****")
            print("커피 매진")
            print("*****")
        else:
            print("*****")
            print("자판기에 잔돈이 부족합니다.")
            print("*****")

    return False

    else:
        return True

def main():
    c = Coffee()
    c.info()
    while True:
        c.request()
        if c.check_amount():
            c.get(c.put_price, c.req_coffee_nums)

```

```

        c.info()
        continue
    else:
        break

# main함수 호출
if __name__ == '__main__':
    main()

```

충분하게 넣은 경우

```

*****
돈을 넣어 주세요: 1000
몇개를 사시겠습니까?: 3
커피를 3개 줍니다
거스름돈은 100 입니다
*****
남은 커피 수: 7
남은 금액: 4100
*****
돈을 넣어 주세요: |

```

반 정도 넣은 경우

```

*****
돈을 넣어 주세요: 500
몇개를 사시겠습니까?: 2
커피를 1개 줍니다
거스름돈은 200 입니다
*****
남은 커피 수: 6
남은 금액: 3800
*****
돈을 넣어 주세요:

```

부족하게 넣은 경우

```
*****
돈을 넣어 주세요: 100
몇개를 사시겠습니까?: 1
[FAIL]돈이 부족하거나 갯수가 이상합니다.
거스름돈은 100 입니다
*****
남은 커피 수: 6
남은 금액: 3800
*****
돈을 넣어 주세요:
```

매진된 경우

```
남은 금액: 2000
*****
돈을 넣어 주세요: 5000
몇개를 사시겠습니까?: 1
*****
커피 매진
*****
```

