



영진전문대학교

글로벌시스템융합과 - GSC

자율학습용 코딩 문제 #4

정영철 교수

글로벌시스템융합과



A.I WITH BETTER LIFE

문제 1 : 가상 주식 거래 시뮬레이션 (1)

- ✓ 당신은 가상 주식 거래 플랫폼을 위한 시뮬레이터를 만들고자 합니다.
- ✓ 사용자는 이 시뮬레이터를 통해 초기 자본금으로 주식을 구매하고,
- ✓ 이후 주식 가격이 변동한 후에 주식을 판매하여 얻는 이익이나 발생하는 손실을 계산할 수 있어야 합니다.

- 사용자는 다음 정보를 **입력**합니다:

- 초기 자본금 (예: 10000)
- 주식의 구매 가격 (예: 100)
- 구매할 주식의 수 (예: 50)
- 판매할 때의 주식 가격 (예: 150)

- 시뮬레이터는 다음을 **계산**해야 합니다:

- 주식 구매 후 남은 자본금
- 주식 판매 시 예상 이익 또는 손실

- 요구 사항

- **주식 구매 비용 계산:** 구매 가격과 주식 수를 곱하여, 총 구매 비용을 계산하세요.
- **남은 자본금 계산:** 초기 자본금에서 총 구매 비용을 빼서, 주식 구매 후 남은 자본금을 계산하세요.
- **판매 예상 이익 계산:**
 - 판매할 때의 주식 가격과 주식 수를 곱하여 총 판매 금액을 계산하고,
 - 여기서 총 구매 비용을 빼서 예상 이익 또는 손실을 계산하세요.
- **결과 출력:** 구매 후 남은 자본금과 예상 이익(또는 손실)을 출력하세요.
 - 이익이 발생했다면 "예상되는 이익입니다."를,
 - 손실이 발생했다면 "예상되는 손실입니다."를 추가로 출력하세요.

문제 1 : 가상 주식 거래 시뮬레이션 (2)

➤ 출력 결과

초기 자본금을 입력하세요 : 1000000
주식 가격을 입력하세요 : 1500
구매할 주식 수를 입력하세요 : 60
판매할 때의 주식 가격을 입력하세요 : 1700
구매 후 남은 자본금 : 910000.00
예상 이익 : 12000.00
예상되는 이익입니다.

초기 자본금을 입력하세요 : 1000000
주식 가격을 입력하세요 : 1500
구매할 주식 수를 입력하세요 : 60
판매할 때의 주식 가격을 입력하세요 : 1000
구매 후 남은 자본금 : 910000.00
예상 이익 : -30000.00
예상되는 손실입니다.

문제 2 : 비밀번호 검증기 : 문제 설명

- 사용자로부터 입력받은 비밀번호가 안전한지를 검증하는 프로그램을 작성.
- 비밀번호가 다음 기준을 모두 충족해야 안전하다고 간주합니다
 1. 비밀번호의 길이가 8자 이상이어야 합니다.
 2. 비밀번호에는 적어도 하나의 숫자가 포함되어 있어야 합니다.
 3. 비밀번호에는 적어도 하나의 대문자가 포함되어 있어야 합니다.
- 위 조건 중 하나라도 충족하지 않는 경우, 비밀번호는 안전하지 않다고 평가해야 합니다.
- 모든 조건이 충족될 때는 "비밀번호가 안전합니다."를, 그렇지 않을 때는 "비밀번호가 안전하지 않습니다."를 출력해야 합니다

➤ 출력 결과

비밀번호를 입력하세요 : 123
비밀번호가 안전하지 않습니다.

비밀번호를 입력하세요 : 12345678
비밀번호가 안전하지 않습니다.

비밀번호를 입력하세요 : abcdefgh
비밀번호가 안전하지 않습니다.

비밀번호를 입력하세요 : abcdefg1
비밀번호가 안전하지 않습니다.

비밀번호를 입력하세요 : abcdefG1
비밀번호가 안전합니다.

문제 2 : 비밀번호 검증기 : 힌트[Hints] (1)

- 문제 해결을 위해 필요한 기본 개념과 함수 사용법에 대한 예제 힌트입니다. 이 힌트를 활용하여, 문제의 요구사항에 맞는 비밀번호 검증 로직을 구현해 보세요.

1. 문자열 길이 확인하기:

- 문자열의 길이를 확인하고자 할 때 **len()** 함수를 사용합니다.
- 예를 들어, 비밀번호의 길이가 충분한지를 확인하는 예제 코드는 다음과 같습니다.

```
2 password = input("비밀번호를 입력하세요: ")
3 if len(password) >= 8:
4     print("비밀번호 길이가 충분합니다.")
5 else:
6     print("비밀번호가 너무 짧습니다.")
```

2. 문자열에서 숫자 확인하기:

- 문자열을 반복하며 각 문자가 숫자인지 확인하려면 **isdigit()** 함수를 사용합니다.
- 비밀번호에 숫자가 포함되어 있는지 확인하는 예제 코드는 다음과 같습니다

```
1 password = "abc3def4g"
2 has_number = False
3 for char in password:
4     if char.isdigit():
5         print(char) # 출력값 : 3
6         has_number = True
7         break # 숫자가 확인되면 더 이상 반복하지 않습니다.
```

문제 2 : 비밀번호 검증기 : 힌트[Hints] (2)

3. 문자열에서 대문자 확인하기:

- 문자열을 반복하며 각 문자가 대문자인지 확인하려면 **isupper()** 함수를 사용합니다.
- 비밀번호에 대문자가 포함되어 있는지 확인하는 예제 코드는 다음과 같습니다

```
1 password = "abc3dEF4G"
2 has_uppercase = False
3 for char in password:
4     if char.isupper():
5         print(char) # 출력값: E
6         has_uppercase = True
7         break # 대문자가 확인되면 더 이상 반복하지 않습니다
```

Q/A

감사합니다



주문식교육의 산실
영진전문대학교