12장

진급 프로젝트:이름 매시업

12장 진급 프로젝트:이름 매시업

12.1 문제에 대한 설명 이해하기

프로젝트 문제

» **이제부터 작성할 프로그램은 사용자가 입력한 성과 이름을 조합한다. 문제에 대한 설명은 다음과 같다. 명확히 정의하지 않은 부분에 대해서는 원하는 대로 세부 사항을 추가하거나 제약을 더 걸어도 좋다.**

- 사용자에게 이름, 성 순서로 성명을 두 개 입력하라고 요청한다.
- 사용자에게 새로운 이름 조합을 이름, 성 순서로 보여준다.
- 새로운 이름은 사용자가 입력한 이름을 조합하고, 새로운 성은 사용자가 입력한 성을 조합해 만든다. 예를 들어 사용자가 Alice Cat과 Bob Dog을 입력하면 Bolice Dot과 같은 이름을 만들 수 있다.

12.1 문제에 대한 설명 이해하기

어떤 문제에 대한 설명을 들은 뒤에는 다음과 같은 요소를 찾아봐야 한다.

- 프로그램이 달성해야 하는 목적에 대한 일반적인 서술
- 사용자에게 입력을 받을지 여부와 입력을 받는다면 어떤 입력을 받아야 할지
- 프로그램이 출력해야 하는 내용
- 다양한 상황에서 프로그램이 어떻게 동작해야 할지

12.1 문제에 대한 설명 이해하기

» **먼저 자신이 잘 사용할 수 있는 방법으로 주어진 과제에 대한 생각을 체계적으로 정리해야 한다.다음 세 가지를 모두 수행하는 것이 이상적이다.**

- 간략한 그림을 그려 요구 받은 내용이 무엇인지 이해한다.
- 작성한 코드를 테스트할 때 쓸 수 있는 예제를 몇 가지 만든다.
- 그림과 예제에서 핵심을 뽑아내 의사 코드(pseudocode)로 추상화한다.

12.1.1 문제를 그림으로 표현하기

- 프로젝트 문제는 사용자에게 입력을 요청한다. 사용자는 이름을 두 가지 입력해야 한다.
- 여러분이만든 프로그램은 입력 받은 문자열을 이름과 성으로 분리한다. 분리한 다음 이름은 이름끼리 조합하고, 성은 성끼리 조합해야 한다.
- 마지막으로 이렇게 조합한 새로운 성과 새로운 이름을 출력한다.

12.1.1 문제를 그림으로 표현하기

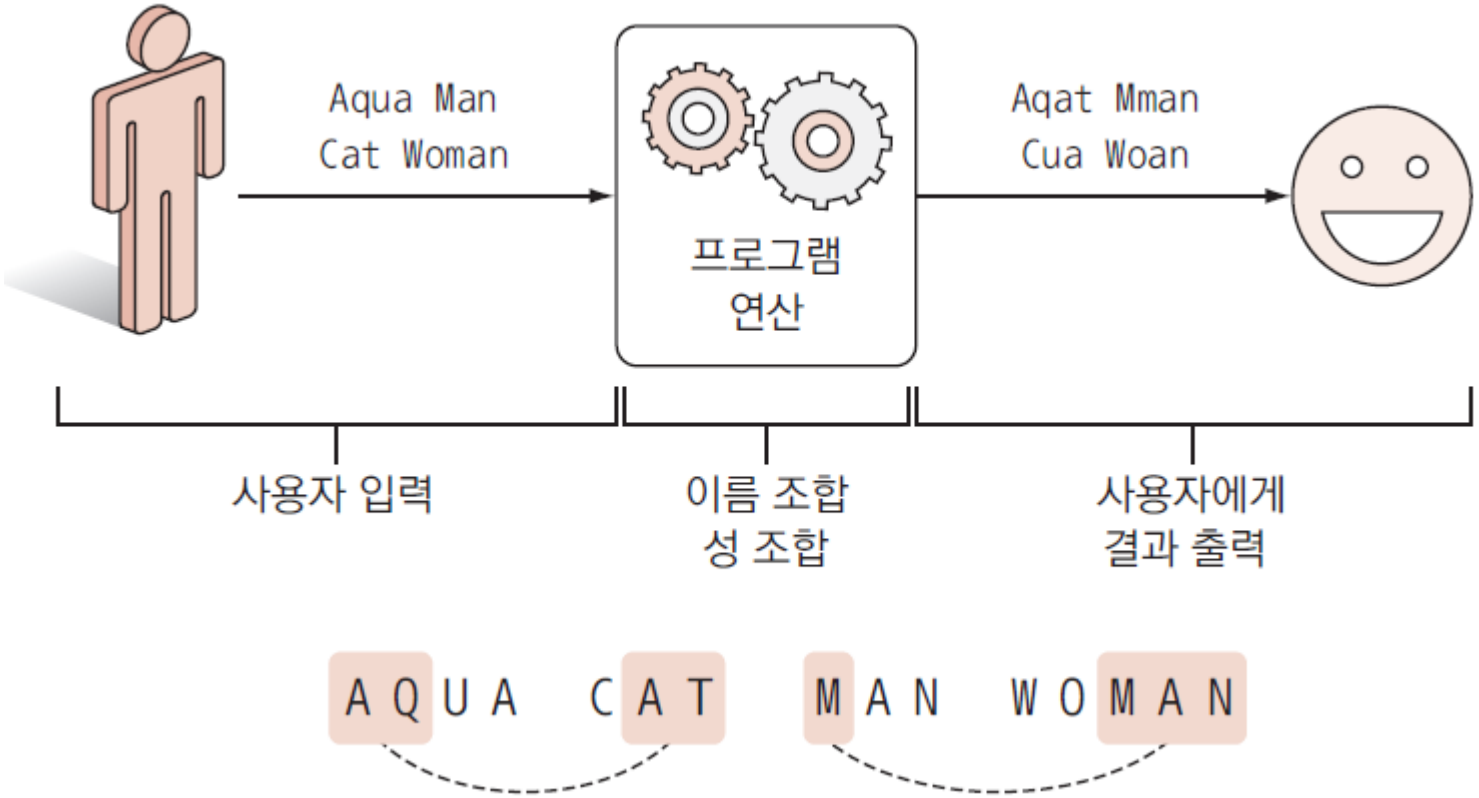


그림 12-1
프로그램의 입력, 입력 조합, 출력 과정을 그린 그림

12.1.2 몇 가지 예제 만들기

테스트에 사용할 수 있는 이름 유형의 예

- 두 글자로 된 이름/성(CJ Cool, AJ Bool)
- 긴 이름/성(Moonandstarsandspace Knight)
- 글자수가 짝수인 이름/성(Lego Hurt)
- 글자수가 홀수인 이름/성(Sting Bling)
- 같은 글자로 이뤄진 이름/성(Aaa Bbb)
- 입력한 두 이름이 같은 경우(Meg Peg와 Meg Peg)

12.1.3 문제를 의사 코드로 추상화하기

» 의사 코드 : 우리말이나 영어 등 일상 언어와 약간의 프로그래밍 구문을 혼합해 이해하기 쉽게 작성한 코드

1. 사용자 입력을 받아서 변수에 저장한다.
2. 전체 이름을 이름과 성으로 분리해서 변수에 각각 저장한다.
3. 이제 두 변수에 있는 이름을 어떻게 잘라서 조합할지 결정한다. 예를 들어 두 이름을 각각 반으로 나눌 수도 있다. 반으로 나눈 앞부분을 각각 변수에 저장하고, 뒷부분도 각각 변수에 저장한다.
4. 한 이름의 앞부분과 다른 이름의 뒷부분을 합친다. 나머지 이름이나 성도 같은 작업을 수행한다

12.2 이름과 성 분리하기

12.2 이름과 성 분리하기

- 지금까지 한 일은 모두 문제가 요구하는 것이 무엇인지 이해하기 위한 것이다. 실수를 저지를 여지를 최소화하려면 코딩은 맨 마지막에 해야 한다.
- 사용자와 상호작용하는 프로그램을 작성할 때는 거의 대부분 사용자 입력을 받는 것부터 시작한다.
- 다음 코드는 입력을 받는 부분이다.

```
print("Welcome to the Mashup Game!")
name1 = input("Enter one full name (FIRST LAST): ")
name2 = input("Enter another full name (FIRST LAST): ")
```

사용자에게 원하는 형식으로 정보를 입력하라고 요청

코드 12-1
산을 가져갈 지 여부를 결정하는 흐름도

- 사용자가 입력한 값을 name1과 name2라는 적절한 이름의 변수에 저장했다.

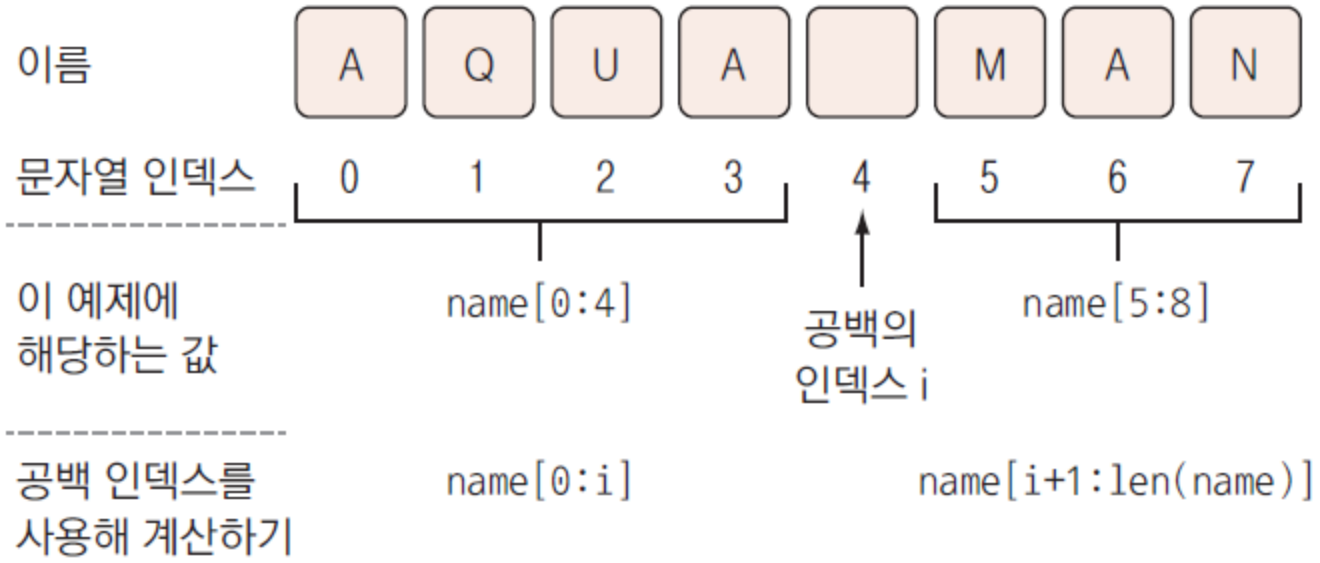
12.2.1 성과 이름 사이의 공백 찾기

전체 이름을 분리하기 위한 첫 단계는 이름과 성 사이에 있는 공백을 찾는 것이다.

- find를 사용하면 문자열에서 어떤 문자의 위치에 해당하는 인덱스를 알 수 있다. 여기서는 공백인 " "의 위치를 찾으면 된다.

12.2.2 변수를 사용해 계산 결과 값 저장하기

- 이제 나중에 사용하기 위해 이름과 성을 변수에 저장해야 한다.
- 우선 공백의 위치 인덱스를 찾는다. 전체 이름의 시작 부분부터 공백까지의 문자는 이름에 해당한 다. 공백 직후부터 문자열 끝까지의 문자는 성에 해당한다.



코드 12-2
전체 이름을 공백 인덱스를 사용해 이름과 성으로 분리하기

12.2.2 변수를 사용해 계산 결과 값 저장하기

- 공백의 인덱스 위치를 알아내면 전체 이름의 시작(인덱스 0)부터 공백 문자의 인덱스 직전까지 있는 문자를 부분 문자열로 취해서 이름에 넣는다.
- 문자열[a:b]라는 연산은 인덱스 a부터 b-1까지모든 문자가 들어 있는 부분 문자열을 돌려준다(그래서 공백이 i 인덱스 위치에 있다면 슬라이스의 : 다음에 i-1이 아니라 i를 넣어야 한다).
- 성을 얻으려면 공백 문자 위치 i 바로 다음부터 전체 이름의 끝(문자열의 마지막에 있는 문자의 인덱스)까지 문자를 가져오면 된다.

12.2.2 변수를 사용해 계산 결과 값 저장하기


```
space = name1.find(" ")
name1_first = name1[0:space]
name1_last = name1[space+1:len(name1)]
space = name2.find(" ")
name2_first = name2[0:space]
name2_last = name2[space+1:len(name2)]
```

---- 공백 문자의 인덱스를 얻어서 저장. 공백 문자로 이름과 성을 구분

---- 첫 문자부터 공백 문자까지(공백은 제외) 모든 문자를 가져와서 이름으로 저장

------ 공백 바로 다음 문자부터 마지막 문자까지 모든 문자를 가져와서 성으로 저장

---- 두 번째 이름에 대해 같은 처리를 반복

코드 12-2
이름과 성을 변수에 넣기

- 여기까지 하고 나면 두 성과 두 이름이 네 변수에 나눠 들어간다.

12.2.3 지금까지 진행한 일 테스트하기

» 여기서는 프로그램출력이 원하는 결과와 일치하는지 반드시 확인해야 한다.

» Aqua Man과 Cat Woman을 입력하고 다음과 같이 print 문으로 결과를 살펴보자.

```
print(name1_first)
print(name1_last)
print(name2_first)
print(name2_last)
```

» 출력은 다음과 같아야 한다

```
Aqua
Man
Cat
Woman
```

12.3 성과 이름을 반으로 나눠 각각 저장하기

12.3 성과 이름을 반으로 나눠 각각 저장하기

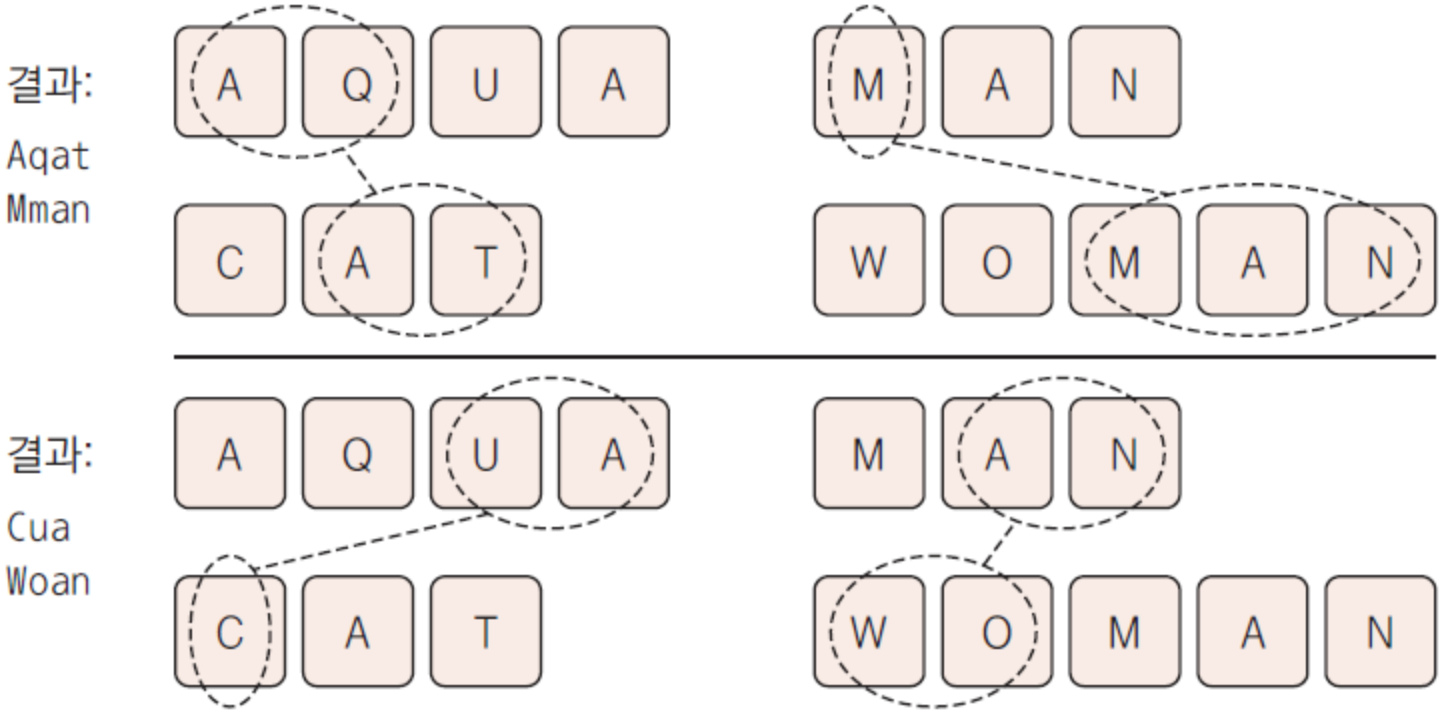


그림 12-3
Aqua Man과 Cat Woman을 조합하는 두 가지 방법. 이름을 반씩 나눠서 서로 교차 조합한 다음, 성도 반씩 나눠서 교차 조합한다

12.3.1 이름의 중간 지점 찾기

- 이름의 중간을 찾으려면 문자열의 중간에 해당하는 인덱스를 계산해야 한다.
- 사용자가 입력한 이름의 길이는 0이 아닌 어떤 양수든 될 수 있고, 그 수는 홀수일 수도 짝수일 수도 있다.

12.3.1.1 이름 길이가 짝수인 경우

- 이름에 포함된 문자 개수가 짝수인 경우는 개수를 반으로 나누기 쉽다.
- 단어의 길이를 구해서2로 나눠 나오는 정수 값이 문자열의 절반에 해당하는 인덱스다.
- 그림1 2-3에서 Aqua가 이런 경우다

12.3.1.2 이름 길이가 홀수인 경우

- 파이썬에서 홀수를 2로 나누면 3.5나 5.5 같은 실수(부동소수점 수)가 나온다. 하지만 실수를 문자열 인덱스로 사용할 수는 없다.
- 실수를 정수로 타입 변환할 수 있다는 사실을 떠올려보자. 예를 들어 int(3.5)는 3과 같다.
- 이런식으로 계산하면 길이가 홀수인 이름의 중간 인덱스는 소수점 이하를 버리기 때문에 앞쪽 절반은 뒤쪽 절반보다 글자 수가 한 글자 적어진다.

12.3.1.3 이름과 성의 절반을 각각 변수에 저장하는 코드

이 코드는 사용자가 입력한 두 성명을 이름과 성으로 나눈 결과를 저장한 네 가지 변수를 처리한다. 두 이름과 두 성의 절반을 찾아내는 이 과정을 네 가지 문자열에 대해 반복하면 된다.

- 우선 이름의 절반 위치를 찾아야 한다.
- 파이썬의 타입 변환 함수를 사용해야 길이가 홀수인 이름을 처리할 수 있다.
- 이름이 5문자라면 앞 절반은 2개, 뒤 절반은 3개의 문자가 들어간다.
- 길이가짝수인 이름은 타입을 변환해도 인덱스가 바뀌지 않는다. 예를 들어 int(3.0)은 3다.

12.3.1.3 이름과 성의 절반을 각각 변수에 저장하는 코드


```
len_name1_first = len(name1_first)
len_name2_first = len(name2_first)
len_name1_last = len(name1_last)
len_name2_last = len(name2_last)
index_name1_first = int(len_name1_first/2)
index_name2_first = int(len_name2_first/2)
index_name1_last = int(len_name1_last/2)
index_name2_last = int(len_name2_last/2)
lefthalf_name1_first = name1_first[0:index_name1_first]
righthalf_name1_first = name1_first[index_name1_first:len_name1_first]
lefthalf_name2_first = name2_first[0:index_name2_first]
righthalf_name2_first = name2_first[index_name2_first:len_name2_first]
lefthalf_name1_last = name1_last[0:index_name1_last]
righthalf_name1_last = name1_last[index_name1_last:len_name1_last]
lefthalf_name2_last = name2_last[0:index_name2_last]
righthalf_name2_last = name2_last[index_name2_last:len_name2_last]
```

입력에서 얻어온 이름과 성의 길이를 저장

길이를 반으로 나눈 값을 정수로 타입 변환하면 소수점 이하를 버림하면서 중간 인덱스를 정수로 계산할 수 있음

앞쪽 절반은 이름의 시작부터 중간 인덱스까지

뒤쪽 절반은 중간 인덱스부터 이름의 끝까지

코드 12-3
이름과 성의 절반 각각 저장하기

12.4 절반의 이름들 조합하기

12.4 절반의 이름들 조합하기

- 절반의 이름들을 조합하려면 관련 변수에 담긴 문자열들을 서로 연결해야 한다. 문자열 연결에는 + 연산자를 사용한다.
- 절반의 이름들을 조합할 때 합쳐진 결과의 첫 글자를 대문자로 만들어서 진짜 이름처럼 보이게 만들어야 한다.
- 코드를 보면 간혹 줄 맨 끝에 역슬래시(\) 기호가 있다.
  - 역슬래시는 코드가 너무 길어서 두 줄로 나눌 때 사용한다.
  - 코드 중간에 줄을 바꾸면서 맨 뒤에 역슬래시를 넣으면 파이썬은 역슬래시가 있는 줄과 다음 줄을 한 줄처럼 처리한다.
  - 역슬래시 없이 프로그램을 실행하면 오류가 발생한다.

12.4 절반의 이름들 조합하기


```
newname1_first = lefthalf_name1_first.capitalize() +  \ ---- 앞쪽 절반에서 첫 글자를 대문자로,
righthalf_name2_first.lower() ---- 뒤쪽 절반 전체를 소문자로 만듦                나머지를 소문자로 만듦
newname1_last = lefthalf_name1_last.capitalize() +  \
righthalf_name2_last.lower()

newname2_first = lefthalf_name2_first.capitalize() +  \
righthalf_name1_first.lower()
newname2_last = lefthalf_name2_last.capitalize() +  \
righthalf_name1_last.lower()

print("All done! Here are two possibilities, pick the one you like best!")
print(newname1_first, newname1_last) ---┐
                                       ├--- 사용자에게 새 이름을 두 가지 보여줌
print(newname2_first, newname2_last) ---┘
```

코드 12-4
이름 조합하기

12.4 절반의 이름들 조합하기

- 먼저 첫 번째 이름의 앞쪽 절반에 capitalize를 적용해 첫 문자를 대문자로, 나머지 문자를 소문자로 만든다.
- 두 번째 이름의 뒤쪽 절반에 lower를 적용해 모든 문자를 소문자로 만든다. 코드가 너무 길기 때문에 역슬래시를 사용해서 긴 코드를 두 줄로 나눠 쓴다.
- 절반의 성과 이름을 각각 합친 후에 마지막으로 해야 할 일은 사용자에게 결과를 보여준다. print 연산을 사용해 새 이름을 출력한다.

12.5 요약

12.5 요약

- 사용자는 프로그램에 입력을 여러 번 할 수 있다.
- find 연산으로 사용자가 입력한 문자열에서 부분 문자열의 위치를 찾을 수 있다.
- 문자열을 변수에 저장한 다음, + 연산을 사용해서 두 문자열을 연결할 수 있다.
- print 연산으로 사용자에게 출력을 보여줄 수 있다