

은행전산시스템

A분반 13조

201411303 이준호

201312428 김정훈

201312439 소경현

Contents

1. 개요	3
2. 용어	3
3. 기본사항	4
3.1 작동환경	4
3.2 프로그램 구성	4
3.3 프로그램 실행	4
4. 프로그램 사용 흐름도	4
5. 데이터요소	4
5.1 날짜	4
5.2 계좌 번호	5
5.3 비밀번호	6
5.4 이름	6
5.5 시간	6
5.6 금액	7
6. 데이터파일	8
6.1 문법 규칙 (형식)	8
6.1.1 사용자 정보	8
6.1.2 계좌 정보	9
6.1.3 거래 내역	11
6.2 해석	12
6.3 의미 규칙(추가 조건)	12
6.3.1 불필요한 중복 방지 규칙	12
6.3.2 식별자 규칙	12
6.4 무결성 검사 및 처리	12
7. 주 프롬프트	13
7.1 주 프롬프트	13
7.1.1 로그인	13
7.1.1.1 부 프롬프트 1 : 계좌 번호 입력	13
7.1.1.2 부 프롬프트 2 : 비밀번호 입력	14
7.1.2 신규 가입	14
7.1.2.1 부 프롬프트 1 : 진입 확인	14
7.1.2.2 부 프롬프트 2 : 이름 입력	15
7.1.2.3 부 프롬프트 3 : 비밀번호 입력	15

7.1.2.4 부 프롬프트 4 : 회원 가입 정보 출력	16
7.1.2.5 부 프롬프트 5 : 가입 확인 메시지 출력	16
7.1.2.6 부 프롬프트 6 : 가입 취소 메시지 출력	16
7.1.3 종료	17
7.2 관리자 메인	17
7.2.1 사용자 정보 조회	17
7.2.2 전체 내역 조회	18
7.2.3 로그아웃	18
7.3 로그인 메인	19
7.3.1 부 프롬프트 1 : 내 정보 확인	19
7.3.2 부 프롬프트 2 : 로그아웃	19
7.4 예금 메인	20
7.4.1 예금 조회	20
7.4.2 내역 조회	21
7.4.2.1 부 프롬프트1 : 날짜 입력1	21
7.4.2.2 부 프롬프트2 : 날짜 입력2	21
7.4.2.3 부 프롬프트3 : 내역 출력	21
7.4.3 입금	22
7.4.3.1 부프롬프트1 : 입금 진입 확인	22
7.4.3.2 부프롬프트2 : 금액 입력	22
7.4.3.3 부 프롬프트3 : 입금 결과 출력	23
7.4.4 출금	23
7.4.4.1 부 프롬프트1 : 출금 단계 진입 확인	23
7.4.4.2 부 프롬프트2: 금액 입력	24
7.4.4.3 부 프롬프트3 : 결과 출력	24
7.4.5 계좌 이체	24
7.4.5.1 부 프롬프트1 : 금액 입력	24
7.4.5.2 부 프롬프트2 : 이체 계좌 입력	25
7.4.5.3 부 프롬프트3 : 이체 확인	25
7.4.5.4 부 프롬프트4 : 이체 결과	26
7.4.5.5 부 프롬프트5 : 이체 취소	26
7.4.6 뒤로 가기	26
7.5 적금 메인	26
7.5.1 적금 조회	27
7.5.2 내역 조회	27
7.5.2.1 부 프롬프트 1 : 시작 날짜	27
7.5.2.2 부 프롬프트 2 : 종료 날짜	28
7.5.2.3 부 프롬프트 3 : 내역 출력	28
7.5.3 입금	29
7.5.3.1 부 프롬프트 1 : 입금 진행 확인	29
7.5.3.2 부 프롬프트 2 : 금액 입력	29
7.5.3.3 부 프롬프트 3 : 입금 결과 출력	30
7.5.4 해약	30

7.5.4.1 부 프롬프트 1: 해약 진행 확인	30
7.5.4.2 부 프롬프트 2: 해약 완료	31
7.5.5 뒤로가기	31

1. 개요

예금, 적금 서비스를 제공하는 은행 정보 시스템

2. 용어

이 문서에서 사용하는 용어들의 의미를 약속합니다.

숫자 표준 키보드로 입력가능한 10개의 아라비아 숫자입니다.

공백 (공백 문자) 표준 키보드에서 스페이스 바로 직접 입력할 수 있는 문자입니다.

홈경로 운영체제에서 제공하는 각 사용자 고유의 홈 디렉토리입니다. 만일 사용자의 계정명이 'test'이고 홈 경로의 설정이 변하지 않았다면 기본적으로 아래와 같습니다.

- Windows10 : "C:\Users\test\Desktop\project2_bank_system"
- OS X : "/Users/test/project2_bank_system"
- Linux / Unix : "/Home/test/project2_bank_system"

개행 (개행 문자) CR 문자(\r), LF문자(\n), CR/LF조합을 통칭합니다.

데이터 파일 은행 시스템이 데이터를 저장하고 읽는 파일로서, 홈경로에 사용자정보, 계좌정보, 거래기록을 담은 파일이 각각 json파일로 저장됩니다.

검색 사용자가 데이터 파일 내의 필요한 정보를 얻기 위해 사용하는 기능입니다.

탭 문자 수평 탭(\t)으로 표준 키보드의 'tab' 버튼과 같은 기능을 합니다.

키 값 검색 시 사용자가 데이터 파일에서 원하는 객체의 값을 가져오기 위해 사용합니다.

Depth 데이터 파일에서 Indentation시 데이터를 오른쪽에 위치한 정도를 수치화한 표현입니다. 파일의 가장 좌측을 depth 1으로 정의합니다.

Indentation 데이터를 쓸 때 문자열을 오른쪽으로 당겨 주변의 문장과 구분하는 것을 의미합니다. python에서는 4개의 space로 정의합니다.

외래 키 어떤 데이터 집합에 소속된 속성이나 속성 집합이 다른 데이터 집합에 기본키가 되는 키를 의미합니다.

비개행공백열 범공백 문자들 중 개행이 아닌 나머지 문자들이 (자유롭게 섞이면서) n번 이상 임의의 갯수만큼 연이어진 나열.

3. 기본사항

3.1 작동환경

표준 Python 3 인터프리터가 설치되어 있는 MS Window 10 상의 Command 창에서 작동합니다.

- Python 2 인터프리터로는 작동하지 않습니다.
- 비표준 Python 3 인터프리터로 작동할 수 있지만, 보장되지는 않습니다.
- Command 창 외에 터미널 인코딩이 유니코드인 다른 터미널 창 (PowerShell, PuTTY, xterm 등)에서도 작동할 수도 있지만, 보장되지는 않습니다.
- 다른 버전의 MS Windows나 Linux / UNIX / OS X에서도 작동할 수도 있지만, 보장되지는 않습니다.

3.2 프로그램 구성

- 프로그램이 배포될 때에는, 주 실행 파일인 bank_main.py 파일과 모듈 파일들, 데이터파일(json)들이 함께 배포됩니다.
- 본인에게 '실행'권한이 있는 경로에 배포된 모든파일을 복사하면 설치됩니다.
- 프로그램이 올바르게 실행되면, 홈 경로에 데이터 파일이 없을 시 생성합니다.

3.3 프로그램 실행

- Command로 실행할 경우, Command 창에서 주 실행 파일을 불러오는 명령어를 사용하여 실행 시킬 수 있습니다.
ex. C:\User\Desktop\project2_bank_system> py bank_main.py

4. 프로그램 사용 흐름도

프로그램 사용 흐름, 즉, 프로그램 사용자 관점에서 프로그램을 사용하는 단계별 경로는 **쪽의 <그림06. 다이어그램01>, <그림07. 다이어그램02>, <그림08. 다이어그램03> 과 같습니다. 프로그램 사용 흐름도는 크게 메인, 예금, 적금 세 부분으로 나누어 표현하였습니다.

5. 데이터요소

데이터 요소에는 날짜, 계좌 번호, 비밀번호, 이름, 시간이 있습니다.

5.1 날짜

날짜는 가입일, 만기일, 내역 조회 시 사용 되는 명칭입니다. 날짜는 그레고리력에 부합해야 한다.

문법 형식 : 문법적으로 올바른 날짜 형식은 아래의 조건 중 하나 이상을 만족하는 문자열입니다.

1. 숫자 2개 + '-' 문자 + 숫자 2개 + '-' 문자 + 숫자 2개 (예: 20-01-01)
2. 숫자 2개 + '/' 문자 + 숫자 2개 + '/' 문자 + 숫자 2개 (예: 20/01/01)
3. 숫자 2개 + '.' 문자 + 숫자 2개 + '.' 문자 + 숫자 2개 (예: 20.01.01)
4. 위의 1 ~ 3 항 중에서 하나 와 맨 앞에 숫자 2개 (예: 2020-01-01 또는 2020/01/01 또는 2020.01.01)
5. 위의 1 ~ 4 항 중에서 '-', '/', '.' 문자들을 제거한 문자열 (예 : 200101 또는 20200101)

기본형 : 숫자 4개 + 숫자 2개 + 숫자 2개 (예시 : 20201231)

동치 비교 : 두 날짜 간의 동치 비교를 할 때에는, 기본형식에서 년, 월, 일에 해당하는 글자가 완전히 일치해야 같은 날짜로 간주합니다.

검색 및 저장 : 입력한 검색어를 이용해 날짜를 검색 또는 저장 할 때 :

- 날짜 검색 시, 문법 형식에 맞는 숫자 두 개를 입력받고 기본형을 바꾸고 두 숫자의 범위 내에 있는 날짜는 매치되는 것으로 간주합니다.
- 문법 형식에서 5항으로 변경 후, 숫자 6자리나 숫자 8자리이어야 합니다.
- 숫자 6자리라면 앞에 두 자리를 년도로 사용합니다.
- 숫자 6자리 중 앞의 두 자리가 40보다 작으면 20을 붙입니다. (24 -> 2024)
- 숫자 6자리 중 앞의 두 자리가 50보다 크면 19를 붙입니다. (65 -> 1965)
- 날짜 검색 및 저장시, 기본형으로 변환하여 사용합니다.

5.2 계좌 번호

예금, 적금 계좌를 식별하는 번호입니다. 특히 예금 계좌번호는 계정 식별번호(ID)로 사용됩니다.

문법 형식 : 문법적으로 올바른 날짜 형식은 아래의 조건 중 하나 이상을 만족하는 문자열입니다.

1. 숫자 4개 + '-' 문자 + 숫자 3개 + '-' 문자 + 숫자 7개 (예: 1234-123-1234567)
2. 숫자 4개 + '/' 문자 + 숫자 3개 + '/' 문자 + 숫자 7개 (예: 1234 /23 1234567)
3. 숫자 4개 + '.' 문자 + 숫자 3개 + '.' 문자 + 숫자 7개 (예: 1234.123.1234567)
4. 위 1 ~ 3 항 중에서 '-', '/', '.' 문자들을 제거한 문자열(예 : 12341231234567)

기본형 : 숫자 14자리 (예 : 12341231234567)

동치 비교 : 두 계좌 번호 간의 동치 비교를 할 때에는, 기본형의 구성과 순서가 일치해야 동일한 계좌 번호로 간주합니다.

검색 및 저장 :

- 계좌번호는 기본형 14자리나 문법형식 1 ~ 3항의 16자리만 허용합니다.
- 계좌번호 검색 및 저장시, 기본형 14자리를 사용합니다.
- 동치 비교 규칙에 부합될 경우 검색 및 저장을 허용합니다.

5.3 비밀 번호

비밀번호는 회원가입시 입력하는 개인정보 입니다. 로그인 시 이용자가 본인의 계정에 접근 권한을 얻기 위해 입력하는 문자열입니다.

문법 형식 : 문법적으로 올바른 비밀번호는 아래의 조건을 모두 만족하는 문자열입니다.

1. 길이가 10 이상 20 이하인 문자열이어야 합니다.
2. 숫자가 1개 이상 포함되어야 합니다. (0-9)
3. 영어 소문자가 1개 이상 포함되어야 합니다. (a-z)
4. 영어 대문자가 1개 이상 포함되어야 합니다. (A-Z)
5. 특수문자가 1개 이상 포함되어야 합니다.
6. 사용 가능한 특수문자 10자: !@#\$%^&*()
7. 숫자, 소문자, 대문자, 10개의 특수문자 이외에는 포함될 수 없습니다.

동치 비교 : 비밀번호의 동치 비교를 할 때에는 두 문자열 전체가 서로 순서와 구성이 완전히 일치해야 같은 비밀번호로 간주합니다.

5.4 이름

이름은 회원가입 시 입력하는 개인정보 입니다. 회원 이름의 정식 명칭입니다.

문법 형식 : 문법적으로 올바른 이름은 아래의 조건을 모두 만족하는 문자열입니다.

1. 길이가 2 이상 10 이하 입니다.
2. 공백을 포함하지 않습니다.
3. 문자열은 모두 한글로 이루어져 있습니다.

동치 비교 : 두 이름의 동치 비교를 할 때에는, 두 문자열 전체가 서로 완전히 일치해야만 같은 이름으로 간주합니다.

검색 및 저장 : 입력한 검색어를 이용해 이름을 찾거나 저장 할 때 :

- 검색어 사이의 공백은 허용하지 않습니다.
- 2 ~ 10자리의 한글만 허용합니다.
- 동치 비교에 부합될 경우 검색 및 저장을 허용합니다.

5.5 시간

시간은 내역 조회 시 날짜와 함께 사용 되는 명칭입니다.

문법 형식 : 문법적으로 올바른 시간 형식은 아래의 조건을 만족하는 문자열입니다.

1. 숫자 2개 + ':' 문자 + 숫자 2개 + ':' 문자 + 숫자 2개 (예: 23:59:59)
2. 24시간 형식을 사용합니다.
3. 처음 숫자 2개는 시간으로 사용하며 00부터 23까지 사용합니다.
4. ':'뒤 숫자는 각각 분, 초로 사용하며 00부터 59까지 사용합니다.

기본형 : 숫자 2개 + 숫자 2개 + 숫자 2개 (예시 : 235959)

동치 비교 : 두 시간의 동치 비교를 할 때에는, 기본형식에서 시, 분, 초에 해당하는 글자가 완전히 일치해야 같은 시간으로 간주합니다.

검색 및 저장 : 입력한 검색어를 이용해 거래 시간 대를 검색할 때,

- 시간 검색 시, 숫자 6자리 두 개를 입력받고 두 숫자의 범위 안에 있는 시간이면 매치된 것으로 간주합니다.
- 시간 저장 시, 기본형으로 변환하여 사용합니다.

5.6 금액

통장 잔고, 이체 금액 등에 사용되는 명칭입니다.

문법 형식 : 문법적으로 올바른 금액 형식은 아래의 조건을 만족하는 문자열입니다.

1. 숫자 1자리 이상 15자리 이하인 음이 아닌 정수를 사용합니다.(0 ~ 999,999,999,999,999)
2. 숫자 2자리 이상의 경우, 첫 번째 숫자는 0이 아니어야 합니다.

동치 비교 : 두 금액의 동치 비교를 할 때에는, 두 문자열이 완전히 일치해야 같은 금액으로 간주합니다.

저장 : 문법 형식에 맞게 저장이 됩니다.

6. 데이터파일

이 프로그램은 은행 정보 시스템 실행 중 사용자로부터 키 입력을 받아들이고 이 정보를 프로그램 내부 동작을 통해 데이터파일에 저장하며, 그와 별개로 사용자가 외부 편집기를 이용해 데이터파일을 직접 편집/저장하는 방식은 지원하지 않습니다. 추가로,

- 6.1의 문법 규칙을 만족하지 않은 경우 정상 작동하지 않습니다.
- 데이터 파일을 직접 편집할 경우, 정상 작동을 보장하지 않습니다.
- 데이터 파일은 UTF-8 인코딩으로 저장되어야 합니다.

이 방식으로 사용자가 데이터 파일을 올바르게 편집할 수 있도록, 그리고 데이터 파일을 어떻게 만들면 어떤 결과가 나올지 정확히 예상할 수 있도록, 데이터 파일의 형식과 조건들을 명시합니다.

6.1 문법 규칙 (형식)

- 문법적으로 올바른 데이터 파일은 key-value 쌍으로 이루어진 json 파일입니다.
- 각 데이터 파일의 indentation을 depth로 정의하고 가장 좌측을 depth 1로 정의할 때, depth의 최대, 최솟값은 아래와 같습니다. 데이터가 존재하지 않을 경우 최소 depth, 데이터가 하나라도 존재할 경우 최대 depth를 만족하며, 이외의 depth는 허용하지 않습니다.

	최소 depth	최대 depth	비고
사용자 정보 데이터	1	3	
계좌 정보 데이터	2	4	Deposits, Savings 키값은 항상 존재.
거래 내역 데이터	1	3	

6.1.1 사용자 정보

은행 정보 시스템 사용자 정보를 가지고 있는 데이터 파일입니다. 하나의 계좌번호는 이름, 비밀번호, 가입일, 계좌 번호 리스트를 포함합니다. 예시는 다음과 같습니다.

```
{
  "12341231234567" : {
    "name" : "김건국",
    "pw" : "Password!",
    "sign_up_date" : "20200930",
    "accounts" : [
      "12341231234567",
      "34511389710192"
    ]
  }
}
```


<그림 01. 사용자 정보 데이터 파일 예시>

- 사용자 정보에 부합하는 데이터 파일 형식 :

```

"<예금 계좌 번호>" : {
  "name" : "<이름>",
  "pw" : "<비밀번호>",
  "sign_up_date" : "<가입일>",
  "accounts" : [
    "<예금 계좌 번호>",
    "<적금 계좌 번호>",
  ]
}
...

```
- <예금 계좌 번호> 는 계좌 번호 데이터를 사용하며 키 값으로 사용됩니다.
- <적금 계좌 번호>는 계좌 번호 데이터를 사용합니다.
- <이름>, <비밀번호>, <가입일>은 각각 이름, 비밀번호, 날짜 데이터를 사용합니다.

6.1.2 계좌 정보

은행 정보 시스템 내의 계좌 세부 정보를 가지고 있는 데이터 파일입니다. 예금과 적금 데이터 파일의 예시는 다음과 같습니다.

```

{
  "Deposits" : {
    "12341231234567" : {
      "balance" : 1500000,
      "sign_up_date" : "20200930"
    },
    "36652126441237" : {
      "balance" : 2000000,
      "sign_up_date" : "20011130"
    }
  },
  "Savings" : {
    "34511389710192" : {
      "balance" : 20000000,
      "sign_up_date" : "20040708",
      "expiration_date" : "20140708"
    },
    "84561389710014" : {
      "balance" : 20000000,
      "sign_up_date" : "19941208",
      "expiration_date" : "20041208"
    }
  }
}

```

<그림 02. 계좌번호 데이터 파일 전체 예시>

```
"Deposits" : {
  "12341231234567" : {
    "balance" : 15000000,
    "sign_in_date" : "20200930"
  },
  "36652126441237" : {
    "balance" : 30000000,
    "sign_in_date" : "20011130"
  }
},
```

<그림 03. 예금 계좌 정보 데이터 파일 예시>

```
"Savings" : {
  "34511389710192" : {
    "balance" : 200000000,
    "sign_in_date" : "20040708",
    "expiration_date" : "20140708"
  },
  "84561389710014" : {
    "balance" : 200000000,
    "sign_in_date" : "19941208",
    "expiration_date" : "20041208"
  }
}
```

<그림 04. 적금 계좌 정보 데이터 파일 예시>

- 예금 계좌 정보에 부합하는 데이터 파일 형식 :
 "Deposits" : {
 "<예금 계좌 번호>" : {
 "balance" : <계좌 잔액>,
 "sign_in_date" : "<가입일>"
 }
 }
- <예금 계좌 번호>는 계좌 번호 데이터를 사용하며 각 계좌 데이터의 키값으로 사용됩니다.
- <계좌 잔액>, <가입일>은 각각 금액 데이터, 날짜 데이터를 사용합니다.
- 적금 계좌 정보에 부합하는 데이터 파일 형식 :
 "Savings" : {
 "<적금 계좌 번호>" : {
 "balance" : <계좌 잔액>,
 "expiration_date" : <만기일>,
 "sign_in_date" : <가입일>
 }
 }

```

    "sign_in_date" : "<가입일>",
    "expiration_date" : "<만기일>"
  }

```

- <적금 계좌 번호>는 계좌 번호 데이터를 사용하며 각 계좌 데이터의 키값으로 사용됩니다.
- <계좌 잔액>은 금액 데이터를 사용하며, <가입일>, <만기일>은 날짜 데이터를 사용합니다.

6.1.3 거래 내역

은행 정보 시스템 내에서 이루어지는 거래 내역을 가지고 있는 데이터 파일입니다. from, to, 거래 날짜, 거래 시간, 거래액을 포함합니다.

```

{
  "12341231234567" : {
    "from" : "12341231234567",
    "to" : "345113897101792",
    "date" : "20200830",
    "time" : "083027",
    "amount" : 1000000
  },
  "345113897101792" : {
    "from" : "12341231234567",
    "to" : "345113897101792",
    "date" : "20200830",
    "time" : "083027",
    "amount" : 1000000
  }
}

```

<그림 05. 거래 내역 데이터 파일 예시>

- 거래 내역 데이터 파일 형식 :
 “<예금 계좌 번호>” : {
 “from” : <송금 계좌>,
 “to” : “<수신 계좌>”,
 “date” : “<거래 날짜>”,
 “time” : “<거래 시간>”,
 “amount” : <거래액>
 }
- <예금 계좌 번호>는 계좌 번호 데이터를 사용하며 해당 사용자 거래 내역 데이터의 키값으로 사용됩니다.
- <송금 계좌>, <수신 계좌>는 계좌 번호 데이터를 사용하며, <거래 날짜>, <거래 시간>, <거래액>은 각각 날짜 데이터, 시간 데이터, 금액 데이터를 사용합니다.

6.2 해석

- 계좌번호의 첫 네 자리가 0~999 범위일 때 관리자 계좌, 1000~4999 범위일 때 예금 계좌, 5000 ~ 9999 범위일 때 적금 계좌로 해석합니다.

6.3 의미 규칙(추가 조건)

6.3.1 불필요한 중복 방지 규칙

- 각 데이터 파일에서 키값으로 사용되는 <예금 계좌 번호>, <적금 계좌 번호>, 사용자의 <이름>, <비밀번호>는 중복되면 안됩니다.
- 이 규칙이 위배될 경우 경고 문구를 출력한 뒤 중복을 제거하는 “경고” 입니다.

6.3.2 식별자 규칙

- 각 데이터 파일의 ‘최소 depth + 1’ 위치의 key값이 특정 데이터를 지목할 수 있는 식별자(identifier)로서의 역할을 합니다. 그 중 각 데이터 파일에서 키값으로 사용되는 <예금 계좌 번호>, <적금 계좌 번호>는 각 데이터 파일을 연결하는 외래 키의 역할을 하기 위해 예금과 적금 계좌번호는 중복되면 안됩니다.
- 이 규칙은 위배될 경우 프로그램이 멈추거나 진행 경로가 아예 달라지는 “오류”입니다.

6.4 무결성 검사 및 처리

프로그램이 실행될 때마다 다음과 같이 데이터 무결성 검사 과정을 거칩니다:

- 홈 경로에 데이터 파일이 있는지 확인해서,
 - 없으면 경고 문구를 출력하면서 홈 경로에 빈 데이터 파일을 새로 생성하고,
 - 있으면 아래 단계로 넘어갑니다.
- 데이터 파일을 처음부터 한 줄 씩 읽으면서 문법을 검사하고 해석한 후 의미 규칙 부합성을 확인합니다. 이때:
 - 문법 규칙에 위배되는 행이 발견되면 상응하는 오류 문구를 출력합니다.
 - 6.3.1절의 불필요한 중복에서 <예금 계좌 번호>, <적금 계좌 번호>의 중복은 무결성 검사에서 발견되지 않습니다. 만약 발견되면 오류 문구를 출력합니다.
 - 6.3.1절의 불필요한 중복에서 <이름>, <비밀번호>의 중복은 무결성 검사에서 발견되지 않습니다. 만약 발견되면 오류 문구를 출력합니다.
 - 6.3.2절의 식별자 충돌이 발견되면 상응하는 오류 문구를 출력합니다.이 과정을 모든 줄에 대해 반복한 후, 아래 단계로 넘어갑니다.
- 위 2번 과정을 모두 거치는 동안, (경고가 아닌) 오류 문구를
 - 한 번이라도 출력했으면 그대로 프로그램을 종료하고
 - 한 번도 출력하지 않았다면 주 프롬프트를 출력하고 사용자의 키 입력을 기다립니다.

7. 주 프롬프트

7.1 주 프롬프트

데이터 무결성 검사를 무사히 통과하면, 화면에 다음과 같은 프롬프트가 출력된 상태입니다 :

은행 전산 시스템 입니다.

1. 로그인
 2. 신규가입
 3. 종료
- >

문법 형식 : 문법적으로 올바른 키 입력 문자열은 숫자 1, 2, 3 중 하나와 앞뒤에 각각 <비개행공백> 이 붙어있는 문자열입니다. 이외의 키 입력 문자열은 잘못된 입력으로 간주하고 [주 프롬프트]가 반복됩니다.

의미 규칙 : 문법적으로만 맞으면, 추가로 꼭 준수해야 할 의미 규칙은 없습니다.

비정상 결과 : 어떤 문자열을 입력하더라도 비정상 결과가 나오지 않습니다.

정상 결과 :

- 숫자 1 : 7.1.1의 로그인 단계로 넘어갑니다.
- 숫자 2 : 7.1.2의 신규 가입 단계로 넘어갑니다.
- 숫자 3 : 프로그램을 종료합니다.

7.1.1 로그인

7.1.1.1 부 프롬프트 1 : 계좌 번호 입력

7.1절 주 프롬프트에서 1번 메뉴를 선택하면 로그인을 위한 계좌번호를 입력하는 부 프롬프트가 출력됩니다:

계좌번호를 입력하세요.

>

문법 형식 : 문법적으로 올바른 키 입력 문자열은 5.2 절의 계좌번호 문법 규칙에 맞는 문자열 앞뒤에 각각 <비개행공백>이 붙어있는 문자열입니다. 즉, 실상 문자가 최소 1개 이상 들어있는 문자열입니다.

해석 : 키 입력 문자열의 맨 앞/뒤에 붙은<비개행공백>들을 모두 제거한 후 나머지를 계좌번호로 해석합니다.

의미 규칙 : 문법적으로만 맞으면, 추가로 꼭 준수해야 할 의미 규칙은 없습니다.

정상 결과 : 7.1.1.2 비밀번호 입력 단계로 넘어갑니다.

7.1.1.2 부 프롬프트 2 : 비밀번호 입력

계좌번호가 입력되면 로그인을 위해 비밀번호를 키 입력하는 부 프롬프트가 출력됩니다:

비밀번호를 입력하세요.

>

문법 형식 : 문법적으로 올바른 키 입력 문자열은 5.3 절의 비밀번호 문법 규칙에 맞는 문자열 앞뒤에 각각 <비개행공백열>이 붙어있는 문자열입니다. 즉, 실상 문자가 최소 1개 이상 들어있는 문자열입니다.

해석 : 키 입력 문자열의 맨 앞/뒤에 붙은<비개행공백열>들을 모두 제거한 후 나머지를 비밀번호로 해석합니다.

의미 규칙 : 문법적으로만 맞으면, 추가로 꼭 준수해야 할 의미 규칙은 없습니다.

비정상 결과 : 인자가 문법 형식에 위배되면 (즉, 인자 자체가 5.3절의 비밀번호 문법 규칙에 맞지 않거나 <비개행공백열>이면) 그에 상응하는 오류 메세 지를 출력하고 이름 입력 부 프롬프트를 다시 출력합니다.

정상 결과 :

- 회원가입 된 일반사용자 : 7.3의 [로그인 메인] 단계로 넘어갑니다.
- 관리자 : 7.2의 [관리자 메인]으로 넘어갑니다.
- 회원가입되지 않은 사용자 : 7.1 [주 프롬프트] 단계로 되돌아갑니다.

7.1.2 신규 가입

주 프롬프트의 2번 메뉴 선택을 통해 새로운 계정을 생성하는 단계입니다. 이름과 비밀번호 입력으로 임의의 계좌번호를 부여받습니다.

7.1.2.1 부 프롬프트 1 : 진입 확인

문법 형식 : 문법적으로 올바른 키 입력 문자열은 영어 알파벳 대문자 Y, N 중 하나와 앞뒤에 각각 <비개행공백열> 이 붙어있는 문자열입니다. 이외의 키 입력 문자열은 잘못된 입력으로 간주하고 진입확인이 반복됩니다.

의미 규칙 : 문법적으로만 맞으면, 추가로 꼭 준수해야 할 의미 규칙은 없습니다.

비정상 결과 : 어떤 문자열을 입력하더라도 비정상 결과가 나오지 않습니다.

정상 결과 :

- Y : 7.1.2.2 의 이름 입력 단계로 넘어갑니다.
- N : 7.1의 [주 프롬프트] 단계로 되돌아갑니다.

7.1.2.2 부 프롬프트 2 : 이름 입력

회원가입 진입이 정상적으로 확인되면 회원 정보를 위한 이름을 키 입력하는 부 프롬프트가 출력됩니다:

**신규 회원 등록입니다.
이름을 입력하세요.
>**

문법 형식 : 문법적으로 올바른 키 입력 문자열은 5.4 절의 이름 문법 규칙에 맞는 문자열 앞뒤에 각각 <비개행공백>이 붙어있는 문자열입니다. 즉, 실상 문자가 최소 1개 이상 들어있는 문자열입니다.

해석 : 키 입력 문자열의 맨 앞/뒤에 붙은<비개행공백>들을 모두 제거한 후 나머지를 이름으로 해석합니다.

의미 규칙 : 문법적으로만 맞으면, 추가로 꼭 준수해야 할 의미 규칙은 없습니다.

비정상 결과 : 인자가 문법 형식에 위배되면 (즉, 인자 자체가 5.4절의 이름 문법 규칙에 맞지 않거나 <비개행공백>이면) 그에 상응하는 오류 메세 지를 출력하고 이름 입력 부 프롬프트를 다시 출력합니다.

정상 결과 :

- 올바른 이름 형식 : 7.1.2.3의 비밀번호 입력 단계로 넘어갑니다.
- 잘못된 이름 형식 : 이름 입력 부 프롬프트 단계로 되돌아갑니다.

7.1.2.3 부 프롬프트 3 : 비밀번호 입력

올바른 형식의 이름을 입력하면 로그인을 위한 비밀번호를 키 입력하는 부 프롬프트가 출력됩니다:

**비밀번호를 입력하세요. (비밀번호는 1 2....)
>**

문법 형식 : 문법적으로 올바른 키 입력 문자열은 5.3 절의 비밀번호 문법 규칙에 맞는 문자열 앞뒤에 각각 <비개행공백>이 붙어있는 문자열입니다. 즉, 실상 문자가 최소 1개 이상 들어있는 문자열입니다.

해석 : 키 입력 문자열의 맨 앞/뒤에 붙은<비개행공백>들을 모두 제거한 후 나머지를 비밀번호로 해석합니다.

의미 규칙 : 문법적으로만 맞으면, 추가로 꼭 준수해야 할 의미 규칙은 없습니다.

비정상 결과 : 인자가 문법 형식에 위배되면 (즉, 인자 자체가 5.3절의 비밀번호 문법 규칙에 맞지 않거나 <비개행공백열>이면) 그에 상응하는 오류 메세 지를 출력하고 이름 입력 부 프롬프트를 다시 출력합니다.

정상 결과 :

- 올바른 비밀번호 형식 : 7.1.2.4의 가입 정보 출력 단계로 넘어갑니다.
- 잘못된 비밀번호 형식 : 비밀번호 입력 부 프롬프트 단계로 되돌아갑니다.

7.1.2.4 부 프롬프트 4 : 회원 가입 정보 출력

올바른 형식의 개인정보를 입력하면 사용자의 회원 가입 정보를 출력하는 부 프롬프트가 출력됩니다:

회원 정보는 다음과 같습니다.

이름 : ...

가입일 : ...

비밀번호 : ...

입력하신 정보가 맞다면 Y, 일치하지 않는다면 N

>

문법 형식 : 이 단계는 문법 규칙도 없고 의미 규칙도 없으므로 어떤 경우에도 비정상 결과를 내놓지 않습니다.

정상 결과 :

- Y : 7.1.2.5의 가입 확인 메시지 출력 단계로 넘어갑니다.
- N : 7.1.2.6의 가입 취소 메시지 출력 단계로 되돌아갑니다.

7.1.2.5 부 프롬프트 5 : 가입 확인 메시지 출력

회원 가입에 동의한다면 가입 완료된 계좌정보를 출력하는 부 프롬프트가 출력됩니다:

가입이 완료되었습니다.

OOO님의 계좌번호는 ... 입니다.

>

문법 형식 : 이 단계는 문법 규칙도 없고 의미 규칙도 없으므로 어떤 경우에도 비정상 결과를 내놓지 않습니다..

정상 결과 :

- 아무 키 : 7.1의 주프롬프트 단계로 돌아갑니다.

7.1.2.6 부 프롬프트 6 : 가입 취소 메시지 출력

회원 가입을 취소한다면 가입 취소 메시지를 출력하는 부 프롬프트가 출력됩니다:

가입이 취소되었습니다.

>

문법 형식 : 이 단계는 문법 규칙도 없고 의미 규칙도 없으므로 어떤 경우에도 비정상 결과를 내놓지 않습니다.

정상 결과 :

- 아무 키 : 7.1의 [주 프롬프트] 단계로 넘어갑니다.

7.1.3 종료

7.1절 [주 프롬프트]에서 3번 메뉴를 선택하면 프로그램이 종료됩니다.

7.2 관리자 메인

관리자 계좌와 비밀번호를 이용해 로그인한 단계입니다. 화면에 다음과 같은 프롬프트가 출력됩니다 :

관리자 입니다.

1. 사용자 정보 조회

2. 전체 내역 조회

3. 로그아웃

>

문법 형식 : 문법적으로 올바른 키 입력 문자열은 숫자 1, 2, 3 중 하나와 앞뒤에 각각 <비개행공백> 이 붙어있는 문자열입니다. 이외의 키 입력 문자열은 잘못된 입력으로 간주하고 [관리자 메인]이 반복됩니다.

의미 규칙 : 문법적으로만 맞으면, 추가로 꼭 준수해야 할 의미 규칙은 없습니다.

비정상 결과 : 어떤 문자열을 입력하더라도 비정상 결과가 나오지 않습니다.

정상 결과 :

- 숫자 1 : 7.2.1의 사용자 정보 조회 단계로 넘어갑니다.
- 숫자 2 : 7.2.2의 전체 내역 조회 단계로 넘어갑니다.
- 숫자 3 : 7.2.3의 로그아웃 단계로 넘어갑니다.

7.2.1 사용자 정보 조회

사용자 정보 조회 단계로 모든 사용자의 <이름>, <가입일>, <예금 계좌번호>, <예금 잔액>, <적금 계좌번호>, <적금 잔액> 을 출력합니다. <이름>은 5.4의 이름 데이터 형식을 이용합니다. <가입일>은 5.1의 날짜 데이터 형식을 이용합니다. <예금 계좌번호>, <적금 계좌번호>는 5.2의 계좌 번호 데이터 형식을 이용합니다. <예금 잔액>, <적금 잔액>은 5.6의 금액 데이터 형식을 이용합니다. 예시는 다음과 같습니다 :

사용자 정보 조회 입니다.

홍길동 / 20200101 / 12341231234567 / 80000 / 56785675678901 / 20000

....

....

총 10명 입니다.

>

문법 형식 : 이 단계는 문법 규칙도 없고 의미 규칙도 없으므로 어떤 경우에도 비정상 결과를 내놓지 않습니다..

정상 결과 :

- 아무 키 : [관리자 메인] 단계로 넘어갑니다.

7.2.2 전체 내역 조회

전체 내역 조회 단계로 모든 거래 내역의 <보낸 계좌>, <받은 계좌>, <거래 날짜>, <거래 시간>, <거래 금액>을 출력합니다. <보낸 계좌>, <받은 계좌>는 5.2의 계좌 번호 데이터 형식을 이용합니다. <거래 날짜>는 5.1의 날짜 데이터 형식을 이용합니다. <거래 시간>은 5.5의 시간 데이터 형식을 이용합니다. <거래 금액>은 5.6의 금액 데이터 형식을 이용합니다. 예시는 다음과 같습니다 :

전체 내역 조회 입니다.

12341231234567 / 56785675678901 / 20200101 / 235959 / 10000

.....

.....

총 20 건 입니다.

문법 형식 : 이 단계는 문법 규칙도 없고 의미 규칙도 없으므로 어떤 경우에도 비정상 결과를 내놓지 않습니다.

정상 결과 :

- 아무 키 : [관리자 메인] 단계로 넘어갑니다.

7.2.3 로그아웃

관리자 계정으로 로그인 되어있던 상태에서 로그아웃합니다.

문법 형식 : 이 단계는 문법 규칙도 없고 의미 규칙도 없으므로 어떤 경우에도 비정상 결과를 내놓지 않습니다..

정상 결과 :

- 아무 키 : [주 프롬프트] 단계로 넘어갑니다.

7.3 로그인 메인

정상적으로 로그인되면 기능 선택을 위해 메뉴를 키 입력하는 부 프롬프트가 출력됩니다:

000님 안녕하세요.

- 1. 예금 조회
 - 2. 적금 조회
 - 3. 내 정보 조회
 - 4. 로그 아웃
- >

문법 형식 : 문법적으로 올바른 키 입력 문자열은 숫자 1, 2, 3, 4 중 하나와 앞뒤에 각각 <비개행공백> 이 붙어있는 문자열입니다. 이외의 키 입력 문자열은 잘못된 입력으로 간주하고 7.3절 [로그인 메인]이 반복됩니다.

의미 규칙 : 문법적으로만 맞으면, 추가로 꼭 준수해야 할 의미 규칙은 없습니다.

비정상 결과 : 어떤 문자열을 입력하더라도 비정상 결과가 나오지 않습니다.

정상 결과 :

- 숫자 1 : 7.4의 [예금 메인] 단계로 넘어갑니다.
- 숫자 2 : 7.5의 [적금 메인] 단계로 넘어갑니다.
- 숫자 3 : 7.3.1의 내 정보 확인 단계 넘어갑니다.
- 숫자 4 : 7.3.2의 주 프롬프트로 돌아갑니다.

7.3.1 부 프롬프트 1 : 내 정보 확인

7.3에서 숫자 3을 입력하면 사용자의 정보를 출력하는 부 프롬프트가 출력됩니다:

000님의 정보입니다.

이름 : ...

가입일 : ...

예금 계좌번호 : ...

적금 계좌번호 : ...

>

문법 형식 : 이 단계는 문법 규칙도 없고 의미 규칙도 없으므로 어떤 경우에도 비정상 결과를 내놓지 않습니다.

정상 결과 : 아무키나 입력하면 7.3절의 [로그인 메인] 단계로 돌아갑니다.

7.3.2 부 프롬프트 2 : 로그아웃

7.3에서 숫자 4를 입력하면 로그아웃합니다.

문법 형식 : 이 단계는 문법 규칙도 없고 의미 규칙도 없으므로 어떤 경우에도 비정상 결과를 내놓지 않습니다.

정상 결과 : 7.1절의 [주 프롬프트]로 돌아갑니다.

7.4 예금 메인

사용자의 계좌번호와 비밀번호를 통해 로그인 후 예금을 선택한 화면입니다. 화면에 다음과 같은 프롬프트가 나타납니다.

예금 메뉴입니다.

1. 예금 조회
 2. 내역 조회
 3. 입금
 4. 출금
 5. 계좌이체
 6. 뒤로가기
- >

문법 형식 : 문법적으로 올바른 키 입력 문자열은 숫자 1, 2, 3, 4, 5, 6 중 하나와 앞뒤에 각각 <비개행공백> 이 붙어있는 문자열입니다. 이외의 키 입력 문자열은 잘못된 입력으로 간주하고 [예금 메인]이 반복됩니다.

의미 규칙 : 문법적으로만 맞으면, 추가로 꼭 준수해야 할 의미 규칙은 없습니다.

비정상 결과 : 어떤 문자열을 입력하더라도 비정상 결과가 나오지 않습니다.

정상 결과 :

- 숫자 1 : 7.4.1의 예금 조회 단계로 넘어갑니다.
- 숫자 2 : 7.4.2의 내역 조회 단계로 넘어갑니다.
- 숫자 3 : 7.4.3의 입금 단계로 넘어갑니다.
- 숫자 4 : 7.4.4의 출금 단계로 넘어갑니다.
- 숫자 5 : 7.4.5의 계좌 이체 단계로 넘어갑니다.
- 숫자 6 : 7.4.6의 [로그인 메인] 단계로 되돌아갑니다.

7.4.1 예금 조회

사용자 예금 조회 단계로 <이름>, <금액> 이 출력됩니다. <이름>은 5.4의 이름 데이터 형식을 이용합니다. <금액>은 5.6의 금액 데이터 형식을 이용합니다. 예시는 다음과 같습니다.

형식통일

예금 조회입니다.

홍길동 님의 잔액은 99999원입니다.

>

문법 형식 : 이 단계는 문법 규칙도 없고 의미 규칙도 없으므로 어떤 경우에도 비정상 결과를 내놓지 않습니다..

정상 결과 :

- 아무 키 : [관리자 메인] 단계로 넘어갑니다.

7.4.2 내역 조회

7.4.2.1 부 프롬프트1 : 날짜 입력1

사용자가 예금 내역을 조회하기 위해 날짜를 입력하는 부 프롬프트가 출력됩니다.

시작 날짜를 입력하세요

>

문법 형식 : 문법적으로 올바른 키 입력은 5.1절의 날짜의 데이터 형식을 따릅니다.

비정상 결과 : 문법적으로 올바른 키입력일 경우 비정상 결과가 출력되지 않습니다.

정상 결과 : 다음 날짜를 입력받는 7.4.2.2의 부 프롬프트2로 넘어갑니다.

7.4.2.2 부 프롬프트2 : 날짜 입력2

사용자가 예금 내역을 조회하기 위해 날짜를 입력하는 부 프롬프트가 출력됩니다.

종료 날짜를 입력하세요

>

문법 형식 : 문법적으로 올바른 키 입력은 5.1절의 날짜의 데이터 형식을 따릅니다.

비정상 결과 : 7.4.2.1에서 입력한 날짜보다 이른 날짜일 경우 오류 메시지를 출력하고 7.4의 [예금 메인]으로 되돌아 갑니다.

정상 결과 : 7.4.2.3의 내역 출력하는 단계로 넘어갑니다.

7.4.2.3 부 프롬프트3 : 내역 출력

사용자 예금 조회 단계로 6.1.3의 거래내역 데이터 파일의 형식을 이용합니다. 거래 내역은 10개씩 출력됩니다. 예시는 다음과 같습니다

예금 거래내역 입니다.

20000000000000 / 20010000000000 / 20201231 / 235959 / 99999

....

....

마지막 페이지입니다. 아무키나 누르면 이전 페이지로 이동합니다.

>

문법 형식 : 이 단계는 문법 규칙이 없습니다. 어떤 경우에도 비정상 결과를 내놓지 않습니다.

정상 결과 :

- 아무 키 :
 - 출력할 내역이 남아 있을 경우 : 다음 내역을 출력합니다.
 - 출력할 내역이 남아 있지 않을 경우 : 7.5의 [예금 메인] 단계로 넘어갑니다.

7.4.3 입금

입금 단계에 진입 시 사용자의 확인을 받은 후 부프롬프트가 출력됩니다.

7.4.3.1 부프롬프트1 : 입금 진입 확인

입금 단계에 진입 시 사용자의 확인을 받습니다.

입금을 계속 진행하시겠습니까?(Y/N)

>

문법 형식 : 문법적으로 올바른 키 입력은 길이 0이상의 모든 문자열입니다.

해석: 정확히 Y 또는 N과 일치해야 하므로 그 외의 모든 입력은 비정상적인 입력으로 간주하고 진입 확인을 반복합니다.

의미 규칙 : 별도의 의미 규칙은 없습니다.

비정상 결과 : 어떠한 경우에도 비정상 결과가 나오지 않습니다.

정상 결과 :

- N : 7.4의 [예금 메인]으로 되돌아갑니다
- Y : 금액을 입력받는 7.4.3.2의 부프롬프트2로 넘어갑니다.

7.4.3.2 부프롬프트2 : 금액 입력

사용자가 입금을 진행할 경우 입금할 <금액>을 입력받는 부프롬프트가 출력됩니다.

금액을 입력하세요.

>

문법형식 : 문법적으로 올바른 키 입력 문자열은 5.6절의 금액 문법 규칙에 맞는 문자열입니다.

의미 규칙 : 문법적으로 맞으면 추가로 꼭 준수해야 할 의미 규칙은 없습니다.

비정상 결과 : 문법적으로 맞으면 비정상 결과가 나오지 않습니다.

정상 결과 : 입금 완료에 해당하는 결과 메시지를 표시합니다.

7.4.3.3 부 프롬프트3 : 입금 결과 출력

입금 결과를 출력합니다. 입금 <금액>과 <잔액>은 5.6의 금액 데이터 형식을 이용합니다. 예시는 다음과 같습니다.

50000원이 입금되었습니다.
잔액은 200000원입니다.
>

문법 형식 : 이 단계는 문법 규칙도 없고 의미 규칙도 없으므로 어떤 경우에도 비정상 결과를 내놓지 않습니다.

정상 결과 :

- 아무 키 : 7.5의 [예금 메인] 단계로 넘어갑니다.

7.4.4 출금

출금 단계에 진입 시 사용자의 확인을 받은 후 부프롬프트가 출력됩니다.

7.4.4.1 부 프롬프트1 : 출금 단계 진입 확인

출금 단계에 진입 시 사용자의 확인을 받습니다.

출금을 계속 진행하시겠습니까?(Y/N)
>

문법 형식 : 문법적으로 올바른 키 입력은 길이 0이상의 모든 문자열입니다.

해석: 정확히 Y 또는 N과 일치해야 하므로 그 외의 모든 입력은 비정상적인 입력으로 간주하고 진입 확인을 반복합니다.

의미 규칙 : 별도의 의미 규칙은 없습니다.

비정상 결과 : 어떠한 경우에도 비정상 결과가 나오지 않습니다.

정상 결과 :

- N : 7.4의 [예금 메인]으로 되돌아갑니다
- Y : 금액을 입력받는 7.4.4.2의 부프롬프트2로 넘어갑니다.

7.4.4.2 부 프롬프트2: 금액 입력

사용자가 출금을 진행할 경우 출금할 <금액>을 입력받는 부프롬프트가 출력됩니다.

금액을 입력하세요.

>

문법 형식 : 문법적으로 올바른 키 입력 문자열은 5.6절의 금액 문법 규칙에 맞는 문자열입니다.

의미 규칙 : 문법적으로 맞으면 추가로 꼭 준수해야 할 의미 규칙은 없습니다.

비정상 결과 : 출금할 금액이 사용자의 예금 계좌의 잔액보다 많으면 상응하는 오류 메시지를 출력하고 [예금 메인] 단계로 이동합니다.

정상 결과 : 출금을 완료에 해당하는 결과 메시지를 출력합니다.

7.4.4.3 부 프롬프트3 : 결과 출력

출금 결과를 출력합니다. 입금 <금액>과 <잔액>은 5.6의 금액 데이터 형식을 이용합니다. 예시는 다음과 같습니다.

50000원이 출금되었습니다.

잔액은 200000원입니다.

>

문법 형식 : 이 단계는 문법 규칙도 없고 의미 규칙도 없으므로 어떤 경우에도 비정상 결과를 내놓지 않습니다.

정상 결과 :

- 아무 키 : 7.5의 [적금 메인] 단계로 넘어갑니다.

7.4.5 계좌 이체

7.4.5.1 부 프롬프트1 : 금액 입력

사용자가 계좌이체를 진행할 경우 이체할 <금액>을 입력받는 부프롬프트가 출력됩니다.

금액을 입력하세요.

>

문법 형식 : 문법적으로 올바른 키 입력 문자열은 5.6절의 금액 문법 규칙에 맞는 문자열입니다.

의미 규칙 : 문법적으로 맞으면 추가로 꼭 준수해야 할 의미 규칙은 없습니다.

비정상 결과 : 출금할 금액이 사용자의 예금 계좌의 잔액보다 많으면 상응하는 오류 메시지를 출력하고 [예금 메인] 단계로 이동합니다.

정상 결과 : 다음 단계인 이체 계좌 입력으로 넘어갑니다.

7.4.5.2 부 프롬프트2 : 이체 계좌 입력

계좌번호를 입력하는 부 프롬프트가 출력됩니다.

계좌번호를 입력하세요.

>

문법 형식 : 문법적으로 올바른 키입력은 5.2절의 <계좌번호> 문법 규칙의 맞는 문자열입니다.

의미 규칙 : 인자가 문법형식에 맞으면 추가로 꼭 준수해야 할 의미 규칙은 없습니다. 단, 현재 로그인된 예금/적금 계좌 번호를 입력하는 것은 허용하지 않습니다.

비정상 결과 : 인자가 문법 형식에 위배되면 그에 상응하는 오류 메시지를 출력하고 [예금 메인] 단계로 이동합니다.

정상 결과 : 다음 단계인 이체 확인으로 넘어갑니다.

7.4.5.3 부 프롬프트3 : 이체 확인

계좌 번호까지 입력하고 나면, 지금까지 입력한 금액, 계좌번호를 다시 출력하고 계좌 이체를 할 것인지 묻습니다.

금액 : 99999

계좌 번호 : 20000001000000

이체하시겠습니까? (Y/N)

>

문법 형식 : 문법적으로 올바른 키 입력은 길이 0이상의 모든 문자열입니다.

해석: 정확히 Y 또는 N과 일치해야 하므로 그 외의 모든 입력은 비정상적인 입력으로 간주하고 이체 확인을 반복합니다.

의미 규칙 : 별도의 의미 규칙은 없습니다.

비정상 결과 : 어떠한 경우에도 비정상 결과가 나오지 않습니다.

정상 결과 :

- N : 지금까지 입력된 금액, 계좌번호 정보를 모두 버립니다.
- Y : 계좌이체를 진행하고, 거래내역 데이터 파일의 맨 끝에 레코드를 저장합니다.
적절한 처리 완료 문구를 출력한 후 [예금 메인]으로 돌아갑니다.

7.4.5.4 부 프롬프트4 : 이체 결과

계좌 이체를 취소한다면 취소 메시지를 출력하는 부 프롬프트가 출력됩니다:

이체가 완료되었습니다.

>

문법 형식 : 이 단계는 문법 규칙도 없고 의미 규칙도 없으므로 어떤 경우에도 비정상 결과를 내놓지 않습니다..

정상 결과 :

- 아무 키 : 7.1의 [주 프롬프트] 단계로 넘어갑니다.

7.4.5.5 부 프롬프트5 : 이체 취소

계좌 이체를 취소한다면 취소 메시지를 출력하는 부 프롬프트가 출력됩니다:

이체가 취소되었습니다.

>

문법 형식 : 이 단계는 문법 규칙도 없고 의미 규칙도 없으므로 어떤 경우에도 비정상 결과를 내놓지 않습니다.

정상 결과 :

- 아무 키 : 7.4의 [예금 메인] 단계로 되돌아갑니다.

7.4.6 뒤로 가기

뒤로 가기를 선택할 경우 [예금 메인]단계가 종료되고 [로그인 메인] 으로 이동합니다.

7.5 적금 메인

적금 단계입니다. 화면에 다음과 같은 프롬프트가 출력됩니다 :

1. 적금 조회
 2. 내역 조회
 3. 입금
 4. 해약
 5. 뒤로가기
- >

문법 형식 : 문법적으로 올바른 키 입력 문자열은 숫자 1, 2, 3, 4, 5 중 하나와 앞뒤에 각각 <비개행공백> 이 붙어있는 문자열입니다. 이외의 키 입력 문자열은 잘못된 입력으로 간주하고 [적금 메인]이 반복됩니다.

의미 규칙 : 문법적으로만 맞으면, 추가로 꼭 준수해야 할 의미 규칙은 없습니다.

비정상 결과 : 어떤 문자열을 입력하더라도 비정상 결과가 나오지 않습니다.

정상 결과 :

- 숫자 1 : 7.5.1의 적금 조회 단계로 넘어갑니다.
- 숫자 2 : 7.5.2의 내역 조회 단계로 넘어갑니다.
- 숫자 3 : 7.5.3의 입금 단계로 넘어갑니다.
- 숫자 4 : 7.5.4의 해약 단계로 넘어갑니다.
- 숫자 5 : 7.3의 [로그인 메인] 단계로 넘어갑니다.

7.5.1 적금 조회

적금 조회 단계로 사용자의 <잔액>, <만기일>을 출력합니다. <만기일>은 5.1의 날짜 데이터 형식을 이용합니다. <잔액>은 5.6의 금액 데이터 형식을 이용합니다. 예시는 다음과 같습니다 :

적금 조회 입니다.
잔액은 50000원 입니다.
만기일은 20201231 입니다.
>

문법 형식 : 이 단계는 문법 규칙도 없고 의미 규칙도 없으므로 어떤 경우에도 비정상 결과를 내놓지 않습니다.

정상 결과 :

- 아무 키 : [적금 메인] 단계로 넘어갑니다.

7.5.2 내역 조회

내역 조회 단계입니다. <시작 날짜>와 <종료 날짜>를 입력하고 해당 기간의 적금 계좌의 내역을 출력합니다. 계좌 내역은 <보낸 계좌>, <받은 계좌>, <거래 날짜>, <거래 시간>, <거래액>으로 이루어져 있습니다. <시작 날짜>, <종료 날짜>, <거래 날짜>는 5.1의 날짜 데이터 형식을 이용합니다. <보낸 계좌>, <받은 계좌>는 5.2의 계좌 번호 데이터 형식을 이용합니다. <거래 시간>은 5.5의 시간 데이터 형식을 이용합니다. <거래 금액>은 5.6의 금액 데이터 형식을 이용합니다.

7.5.2.1 부 프롬프트 1 : 시작 날짜

시작 날짜 입력 단계입니다. 내역 조회시 시작 날짜로 사용됩니다. 예시는 다음과 같습니다.

내역 조회 입니다.
시작 날짜를 입력하세요

>

문법 형식 : 문법적으로 올바른 키 입력 문자열은 날짜 데이터 형식을 따릅니다.

의미 규칙 : 시작 날짜는 종료 날짜 보다 빠르거나 같아야 합니다.

비정상 결과 : 문법이나 의미 규칙에 부합하지 않는 경우 오류로 판단해 비정상 결과를 출력합니다. 비정상 결과 출력은 다음과 같습니다:

올바르지 않은 날짜 입력입니다.

>

비정상 결과 출력 후 아무 키를 누르면 [적금 메인]으로 돌아갑니다.

정상 결과 : 7.5.2.2의 종료 날짜 단계로 넘어갑니다.

7.5.2.2 부 프롬프트 2 : 종료 날짜

종료 날짜 입력 단계입니다. 내역 조회시 종료 날짜로 사용됩니다. 예시는 다음과 같습니다.

종료 날짜를 입력하세요

>

문법 형식 : 문법적으로 올바른 키 입력 문자열은 날짜 데이터 형식을 따릅니다.

의미 규칙 : 시작 날짜는 종료 날짜 보다 빠르거나 같아야 합니다.

비정상 결과 : 문법이나 의미 규칙에 부합하지 않는 경우 오류로 판단해 비정상 결과를 출력합니다. 비정상 결과 출력은 다음과 같습니다:

올바르지 않은 날짜 입력입니다.

비정상 결과 출력 후 아무 키를 누르면 [적금 메인]으로 돌아갑니다.

정상 결과 : 7.5.2.3의 내역 출력 단계로 넘어갑니다.

7.5.2.3 부 프롬프트 3 : 내역 출력

내역 출력 단계입니다. 시작 날짜부터 종료 날짜까지의 사용자 적금계좌의 모든 거래 내역을 출력합니다. 거래 내역은 10개씩 출력되며 출력 내용이 없을 때까지 반복됩니다. 예시는 다음과 같습니다:

홍길동님의 내역입니다.

12341231234567 / 56785675678901 / 20200101 / 235959 / 20000

....

....

....
마지막 페이지 입니다.
>

문법 형식 : 이 단계는 문법 규칙도 없고 의미 규칙도 없으므로 어떤 경우에도 비정상 결과를 내놓지 않습니다..

정상 결과 :

- 아무 키 :
 - 출력할 내역이 남아 있을 경우 : 다음 내역을 출력합니다.
 - 출력할 내역이 남아 있지 않을 경우 : 7.5의 [적금 메인] 단계로 넘어갑니다.

7.5.3 입금

입금 단계입니다. <거래 금액>을 입력하고 금액을 입금합니다. <거래 금액>은 5.6의 금액 데이터 형식을 이용합니다.

7.5.3.1 부 프롬프트 1 : 입금 진행 확인

입금을 진행하는지 확인합니다. 예시는 다음과 같습니다.

입금을 진행하시겠습니까?
>

문법 형식 : 문법적으로 올바른 키 입력 문자열은 영어 알파벳 대문자 Y, N 중 하나와 앞뒤에 각각 <비개행공백열> 이 붙어있는 문자열입니다. 이외의 키 입력 문자열은 잘못된 입력으로 간주하고 진입확인이 반복됩니다.

의미 규칙 : 문법적으로만 맞으면, 추가로 꼭 준수해야 할 의미 규칙은 없습니다.

비정상 결과 : 어떤 문자열을 입력하더라도 비정상 결과가 나오지 않습니다.

정상 결과 :

- Y : 7.5.3.2의 금액 입력 단계로 넘어갑니다.
- N : 7.5의 [적금 메인] 단계로 되돌아갑니다.

7.5.3.2 부 프롬프트 2 : 금액 입력

적금 계좌에 입금할 금액을 입력합니다. 예시는 다음과 같습니다.

금액을 입력하세요
>

문법 형식 : 문법적으로 올바른 키 입력 문자열은 금액 데이터 형식에 부합하고 앞뒤에 각각 <비개행공백열> 이 0개 이상 붙어있는 문자열입니다.

의미 규칙 : 문법적으로만 맞으면, 추가로 꼭 준수해야 할 의미 규칙은 없습니다.

비정상 결과 : 금액 데이터 형식에 부합하지 않는 입력은 비정상 결과를 출력합니다. 비정상 결과는 다음과 같습니다 :

금액이 올바르지 않습니다.

이후 아무 키를 누르면 7.5의 [적금 메인]으로 넘어갑니다.

정상 결과 : 7.5.3.3의 입금 결과 출력 으로 넘어갑니다.

7.5.3.3 부 프롬프트 3 : 입금 결과 출력

입금 결과를 출력합니다. 입금 <금액>과 <잔액>은 5.6의 금액 데이터 형식을 이용합니다. 예시는 다음과 같습니다.

50000원이 입금되었습니다.

잔액은 200000원입니다.

>

문법 형식 : 이 단계는 문법 규칙도 없고 의미 규칙도 없으므로 어떤 경우에도 비정상 결과를 내놓지 않습니다.

정상 결과 :

- 아무 키 : 7.5의 [적금 메인] 단계로 넘어갑니다.

7.5.4 해약

적금 해약을 진행합니다. 해약시 적금은 이율을 보장하지 않으며 금액은 예금 계좌로 이체되고, 적금 계좌는 삭제됩니다.

7.5.4.1 부 프롬프트 1: 해약 진행 확인

해약을 진행하는지 확인합니다. 예시는 다음과 같습니다.

해약을 진행하시겠습니까?

>

문법 형식 : 문법적으로 올바른 키 입력 문자열은 영어 알파벳 대문자 Y, N 중 하나와 앞뒤에 각각 <비개행공백> 이 붙어있는 문자열입니다. 이외의 키 입력 문자열은 잘못된 입력으로 간주하고 진입확인이 반복됩니다.

의미 규칙 : 문법적으로만 맞으면, 추가로 꼭 준수해야 할 의미 규칙은 없습니다.

비정상 결과 : 어떤 문자열을 입력하더라도 비정상 결과가 나오지 않습니다.

정상 결과 :

- Y : 7.5.4.2의 해약 완료 단계로 넘어갑니다.
- N : 7.5의 [적금 메인] 단계로 되돌아갑니다.

7.5.4.2 부 프롬프트 2 : 해약 완료

해약 결과를 출력합니다. <금액>은 5.6의 금액 데이터 형식을 이용합니다. 예시는 다음과 같습니다.

```
해약이 완료 되었습니다.  
적금 해약 금액은 100000원 입니다.  
>
```

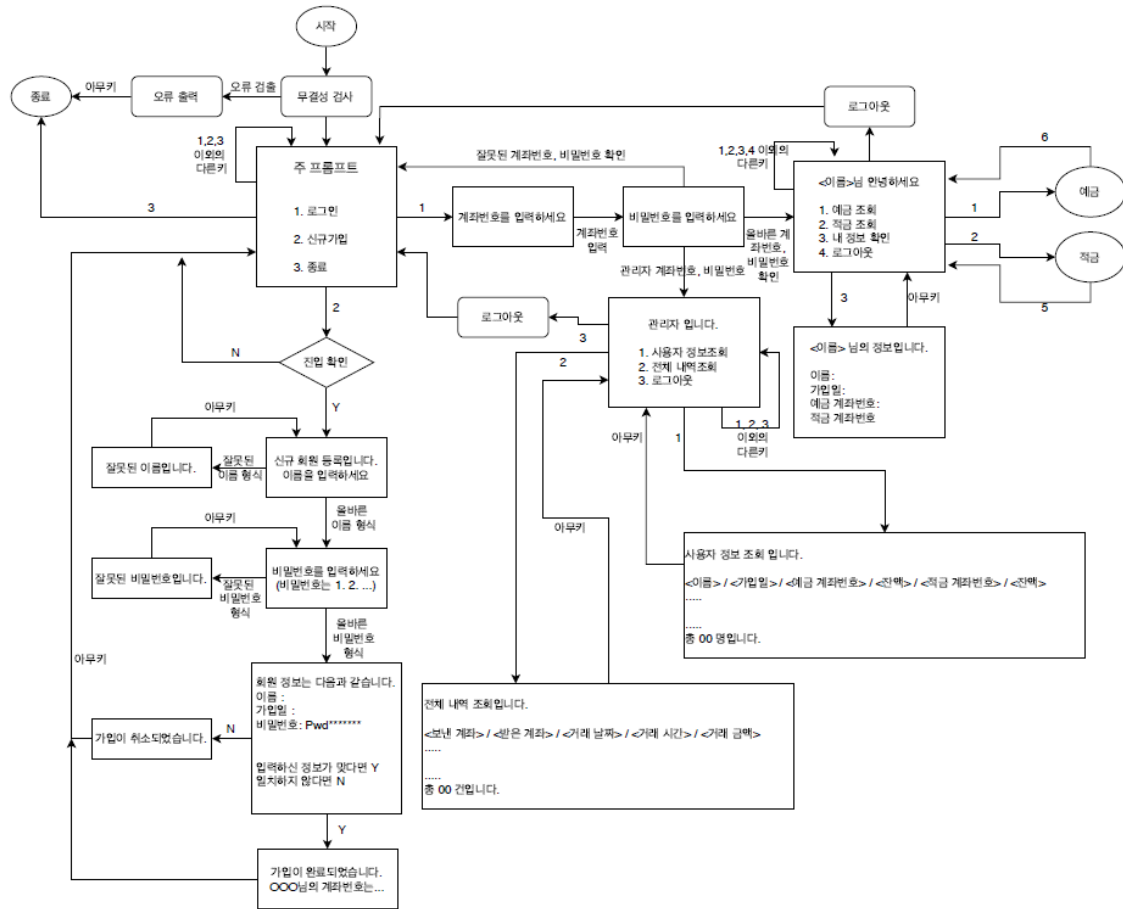
문법 형식 : 이 단계는 문법 규칙도 없고 의미 규칙도 없으므로 어떤 경우에도 비정상 결과를 내놓지 않습니다.

정상 결과 :

- 아무 키 : 7.5의 [적금 메인] 단계로 넘어갑니다.

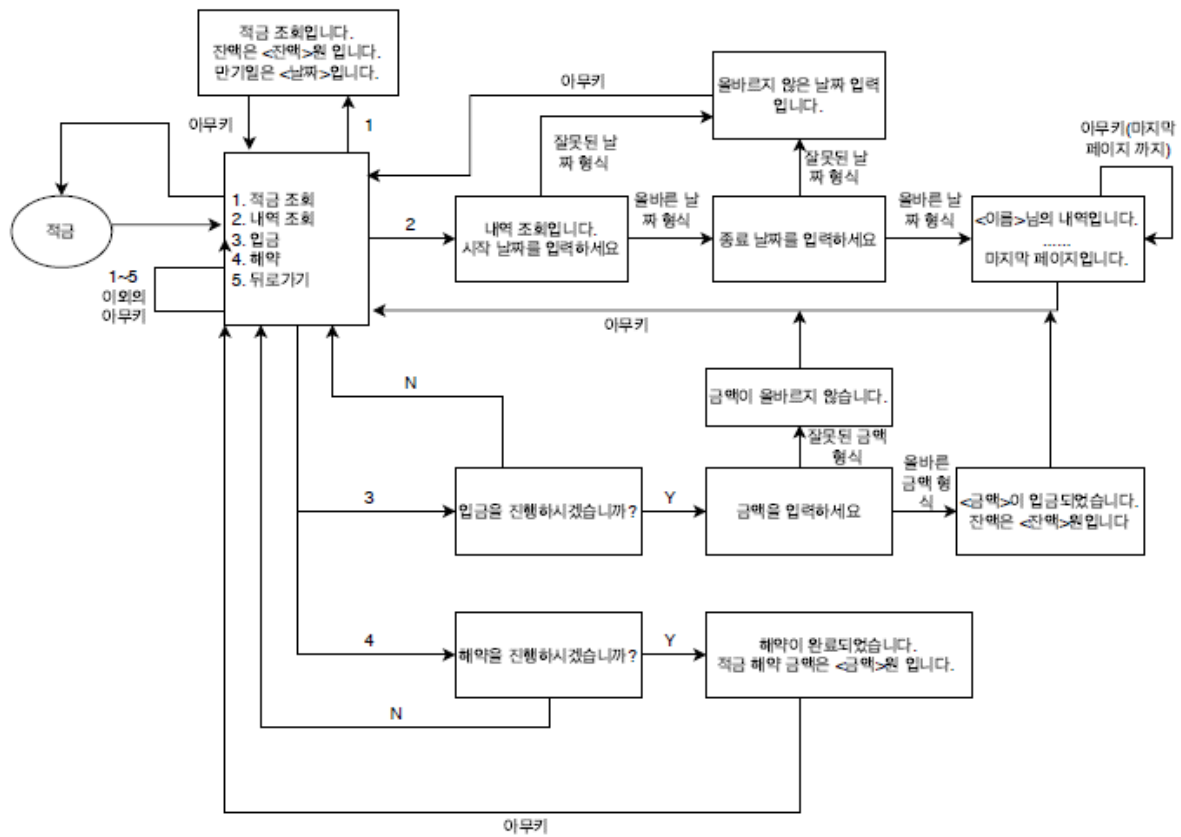
7.5.5 뒤로가기

뒤로 가기를 선택할 경우 [적금 메인] 단계가 종료되고 [로그인 메인] 으로 이동합니다.



<그림 06. 다이어그램 01>

33/34



<그림 08. 다이어그램 03>