**편의점 POS기**

**1차 기획서**

A분반 1조

201911144 강건우

201911146 강병우

201911182 성지운

201911194 유광채

201911215 조재현

**Contents**

1. **개요……………………………………………………………………………………………** 3
2. **용어 ………………………………………………………………………………………………** 3
3. **기본사항**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .4

3.1 작동 환경. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .4

3.2 프로그램 구성 및 설치. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .4

3.3 프로그램 실행. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .5

1. **프로그램 사용 흐름도**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
2. **데이터 요소**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .5

5.1 상품 코드. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

5.2 결제 코드. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

5.3 상품명. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .6

5.4 카테고리 명. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

5.5 카테고리 코드. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .7

5.6 세부 카테고리 코드. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

5.7 가격. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .8

5.8 날짜. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .8

5.9 유통기한 설정 값. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .9

5.10 수량. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

1. **데이터 파일**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .10
2. **주 프롬프트** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .10

7.1 결제 명령 ‘1’ : . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .12

7.1.1 부 프롬프트 1 : 상품코드 입력. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

7.1.2 부 프롬프트 2 : 결제수단 선택 명령. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .12

7.1.3 부 프롬프트 3 : 결제할지 묻기. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .13

7.1.4 부 프롬프트 4 : 금액 입력(현금) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .13

7.2 환불 명령 ‘2’ : . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .14

7.2.1 부 프롬프트 1 : 결제 코드 입력. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .14

7.2.2 부 프롬프트 1 : 상품코드 입력. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .14

7.2.3 부 프롬프트 2 : 환불여부 묻기. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

7.3 재고관리 명령 ‘3’ : . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

7.3.1 부 프롬프트 1 : 재고 관리 세부 명령 선택 . . . . . . . . . . . . . . . . . . . . . . . . . .15

7.3.2 부 프롬프트 2.1 : 상품명 입력. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .16

7.3.3 부 프롬프트 2.2 : 카테고리명 입력. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .16

7.3.4 부 프롬프트 2.3 : 유통기한 입력. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .17

7.3.5 부 프롬프트 2.4 : 가격입력. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

7.3.6 부 프롬프트 2.5 : 상품정보를 추가할지 묻기. . . . . . . . . . . . . . . . . . . . . . . . . .17

7.3.7 부 프롬프트 3.1 : 상품 재고 추가. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

7.3.8 부 프롬프트 3.2 : 상품 재고 수량 입력. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .18

7.3.9 부 프롬프트 4.1 : 폐기할 상품코드 입력. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

7.3.10 부 프롬프트 4.2 : 폐기할지 묻기. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .19

7.3.11 부 프롬프트 5 : 종료. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .20

7.4 현금 충전 명령 ‘4’: . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .20

7.4.1 부 프롬프트 1 : 현금 충전 세부 명령 선택. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

7.4.2 부 프롬프트 2 : 수량 입력. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .21

7.4.3 부 프롬프트 3 : 충전할지 묻기. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

7.5 매출 조회 명령 ‘5’ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .22

7.5.1 부 프롬프트 : 매출 조회. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

7.6 종료 명령 ‘6’ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .22

**1. 개요**

편의점 POS기를 구현한 프로그램

**2. 용어**

이 문서에서 사용되는 용어들의 의미는 다음 정의와 같습니다 :

**결제 리스트 -** 결제한 내역별로 구매한 상품들을 저장하고 있는 텍스트파일.

**상품명 -** 각각의 상품이 가진 고유한 이름

**상품코드 -** 각 상품마다 부여되는 6자리의 고유한 코드. 결제와 환불, 재고관리에 상품 탐색을 위해 사용되는 코드.

**결제코드 -** 각각의 거래를 저장, 관리하기위해 발급되는 9자리의 코드. 환불시 사용된다.

**카테고리 명 -** 상품을 분류하기 위해 설정하는 포괄적인 범위의 범주.

**카테고리 코드 -** 카테고리명에 대응하는 코드, 상품 코드에 생성에 사용된다.

**세부 카테고리 코드 -** 세부 카테고리명에 대응하는 코드, 상품 코드에 생성에 사용된다.

**데이터 파일 -** POS기 프로그램이 상품과 데이터를 저장하고 읽는 텍스트 파일로서, 각 카테고리 코드 2자리를 이름으로 한 텍스트파일과 결제 리스트, 그리고 날짜 텍스트파일이 존재.

**(문자열의) 길이 -** 문자열의 길이는 항상 그 문자열 속에 들어있는 문자의 실제 갯수만 세며 (단위 : 문자, 글자, 자, 개 등), 메모리 상의 용량이나 글꼴 폭 정보 (wide, narrow 등)와는 전혀 무관합니다.

**날짜 -** 윤년을 적용하지 않고, 1년은 항상 365일이고, 2월은 28일까지, 4, 6, 9, 11월은 30일까지, 나머지 달은 31일까지 있습니다.

**메인메뉴 -** 프로그램 실행 후 날짜를 정상적으로 입력했을 때 화면에 출력되어 입력을 받는 과정.

**메뉴 선택방식 -** 메인메뉴 과정에서 1~6 중 하나를 입력해 명령을 내리는 방식.

**Y/N형식 -** Y 또는 N을 입력하는 형식으로, 입력한 값에 따라 분기가 나뉩니다.

**3. 기본사항**

**3.1 작동환경**

JDK 1.8이 설치되어있는 MS Windows 10 상의 환경에서 작동합니다.

* 다른 JDK 버전에서는 정상적으로 작동하지 않을 수 있습니다.
* Eclipse JAVA 2020-09 버전에서 개발되어 다른 버전이나 다른 IDE에서는 작동할 수 있지만, 보장되지는 않습니다.
* 다른 버전의 MS Windows나 다른 운영체제에서도 작동할 수도 있지만, 보장되지는 않습니다.

**3.2 프로그램 구성 및 설치**

* 프로그램이 배포될 때에는, 주 실행파일인 main.java파일이 포함된 패키지가 압축되어 배포됩니다.
* 본인에게 ‘실행’ 권한이 있는 경로에 프로그램 패키지를 복사하면 설치됩니다.
* 프로그램이 실행되면서 없던 txt파일이 생성될 수 있습니다.
* txt 파일은 7쪽의 6. 데이터 파일의 파일과 같습니다.

**3.3 프로그램 실행**

* cmd창에서 실행할 경우, cd 명령어를 통해 배포된 패키지가 설치된 경로로 이동합니다.
* 이후 javac 명령어 뒤에 패키지명(=패키지경로)과 주 실행파일 이름을 입력해 컴파일을 한 후 java 명령어 뒤에 패키지명(=패키지경로)과 주 실행파일 이름을 입력해 실행시킬 수 있습니다.

예) cd C:\Users\zim25\Desktop

javac sw/project/pos/main.java

java sw/project/pos/main

패키지명은 아직 미정이기 때문에 예시에 나온 패키지명과 다른 이름으로 배포될 수 있습니다.

**4. 프로그램 사용 흐름도**

프로그램 사용 흐름, 즉, 프로그램 사용자 관점에서 프로그램을 사용하는 단계별 경로는 26쪽의 “Figure 1:

사용 흐름도”와 같습니다.

**5. 데이터 요소**

데이터 요소에는 크게 상품코드, 결제코드, 상품명, 카테고리 명, 카테고리 코드, 세부 카테고리 명, 세부 카테고리 코드, 가격, 날짜(포스기 최초 입력값, 상품 등록시 입력값), 유통기한 설정 값, 수량이 있습니다.

**5.1 상품 코드**

상품 코드는 각각의 상품에 부여된 고유한 코드 입니다.

**문법 형식 :** 문법적으로 올바른 상품 코드는 아래 4가지 조건을 만족해야합니다.

* 길이가 6을 만족하면서
* 각 자리의 문자는 모두 로마자 대문자를 만족하고
* 특수 문자 또는 공백이 없는 문자.
* 아래의 정규표현식을 만족합니다

<상품 코드> ::= [A-Z]{6}

**생성 규칙 :** 길이가 6인 문자열로 생성되며 앞에서부터 2자리는 카테고리의 코드, 다음의 2자리는 세부 카테고리의 코드, 마지막 2자리는 동일 카테고리와 동일 세부 카테고리를 가진 상품들 중 재고에 추가된 순서를 각각 AA부터 ZZ까지 순서대로 추가. 생성할 수 있는 모든 경우를 사용하면 더 이상 생성할 수 없습니다.

**동치 비교 :** 두 상품코드간의 동치 비교를 할 때는, 소문자가 있다면 대문자로 전환해주고,

두 문자열 전체가 서로 완전히 일치해야만 같은 상품 코드로 간주합니다.

**검색 :** 입력한 검색어를 상품 코드중에서 찾을때에는:

* 검색어와 상품 코드 모두 로마자 대문자로 치환하고
* 검색어가 상품 코드의 문자열과 완전히 같다면, 매치된 것으로 간주합니다.

**5.2 결제 코드**

결제 코드는 결제가 완료된 후 해당 거래의 기록을 나타내는 고유한 코드입니다.

**문법 형식 :** 문법적으로 올바른 결제 코드는 아래 조건을 모두 만족시키는 문자열입니다.

* 길이가 9를 만족하면서
* 각 자리의 문자는 아라비아 숫자여야 합니다.
* 특수 문자 또는 공백이 없는 문자.
* 아래의 정규표현식을 만족합니다

<결제코드> ::= [0-9]{2}( 02(0[1-9] | 1[0-9] | 2[0-8] ) | ( 04 | 06 | 09 | 11 )( 0[1-9] | [12][0-9] | 30 ) | ( 01 | 03 | 05 | 07 | 08 | 10 | 12 )( 0[1-9] | [12][0-9] | 3[01] ))( 00[1-9] | 0[1-9][0-9] | [1-9][0-9][0-9] )

**생성 규칙 :** 길이가 9인 문자열로 생성되며 앞에서부터 6자리는 결제한 날의 4자리 년도 뒤의 2자리, 월 2자리, 일 2자리이고, 마지막 3자리는 동일 날짜에 결제한 순서대로 각각 001부터 999까지 순서대로 추가.생성할 수 있는 모든 경우를 사용하면 더 이상 결제할 수 없습니다.

**동치 비교 :** 두 결제 코드 간의 동치 비교를 할 때는, 두 문자열 전체가 서로 완전히 일치해야만 같은 결제 코드로 간주합니다.

**검색**: 입력한 검색어를 결제 코드중에서 찾을때에는

* 검색어와 결제 코드의 문자열과 완전히 같아야만, 매치된 것으로 간주합니다.

**5.3 상품명**

상품명은 각각의 상품이 가진 고유한 이름입니다.

**문법 형식 :** 문법적으로 올바른 상품명은 아래 조건들을 모두 만족시키는 문자열입니다:

* 길이가 1 이상 (즉, 빈 문자열은 불가능) 10 이하이고,
* 공백이 아닌 문자들을 의미하며,
* 한글 이외의 언어는 허용하지 않습니다.
* 정규표현식으로 표현하면 다음과 같습니다.

<상품명> ::= ( [가-힣] | [ㄱ-ㅎ] | [ㅏ-ㅣ] ){1,10}

**동치 비교 :** 두 상품명 간의 동치 비교를 할 때에는, 두 문자열 전체가 완전히 동일한 한글로 이루어진 문자열인 경우 동일한 것으로 간주합니다.

**검색**: 입력한 검색어를 상품명 중에서 찾을때에는

* 검색어와 상품명의 문자열과 완전히 같아야만, 매치된 것으로 간주합니다.

**5.4 카테고리 명**

카테고리명은 상품을 분류하는 범주에 붙는 이름입니다.

**문법 형식:** 문법적으로 올바른 카테고리명은 아래 조건에 모두 부합하는 문자열입니다.

* 길이가 1 이상(즉, 빈 문자열은 불가능) 10이하 이고,
* 공백이 아닌 문자들을 의미하며
* 한글 이외의 언어는 허용하지 않습니다.
* 정규표현식으로 표현하면 다음과 같습니다.

<상품명> ::= ( [가-힣] | [ㄱ-ㅎ] | [ㅏ-ㅣ] ){1,10}

**동치 비교 :** 두 카테고리 명 간의 동치 비교를 할 때에는, 두 문자열 전체가 완전히 동일한 한글로 이루어진 문자열인 경우 동일한 것으로 간주합니다.

**검색:** 두 카테고리명을 비교할 때에는, 두 문자열 전체가 완전히 동일한 한글로 이루어진 문자열인 경우 동일한 것으로 간주합니다.

**5.5 카테고리 코드**

카테고리는 상품을 분류하기 위해 설정하는 포괄적인 범위의 범주입니다. 676개 초과의 갯수는 만들어지지 않습니다. 만들 수 있는 카테고리 수를 초과한 경우, 새로운 카테고리 생성이 불가합니다.

**문법 형식 :** 문법적으로 올바른 카테고리는 아래 4가지 조건을 만족해야합니다.

* 길이가 2를 만족하면서
* 각 자리의 문자는 로마자 대문자를 만족하고
* 특수 문자 또는 공백이 없는 문자.
* 아래의 정규표현식을 만족합니다

<카테고리> ::= [A-Z]{2}

**생성규칙:** 길이가 2인 문자열로 생성되며, 다른 카테고리 명이 없을 경우 프로그램에 등록된 순서로 각각 AA부터 ZZ까지 순서대로 추가. 생성할 수 있는 모든 경우를 사용하면 더 이상 생성할 수 없습니다.

**동치 비교 :** 두 카테고리 코드 간의 동치 비교를 할 때에는, 두 문자열 전체가 완전히 동일한 로마자 대문자로 이루어진 문자열인 경우 동일한 것으로 간주합니다.

**검색 :** 입력한 검색어를 카테고리 코드 중에서 찾을때에는:

* 검색어와 카테고리 코드 모두 로마자 대문자로 치환하고
* 검색어가 카테고리 코드의 문자열과 완전히 같다면, 매치된 것으로 간주합니다.

**5.6 세부 카테고리 코드**

세부 카테고리는 같은 카테고리에 포함되는 상품들을 더욱 세분화하기 위해 붙이는 구체적인 범주입니다. 676개 초과의 갯수는 만들어지지 않습니다. 만들 수 있는 세부 카테고리 수를 초과한 경우, 새로운 세부 카테고리 생성이 불가합니다.

**문법 형식 :** 문법적으로 올바른 세부 카테고리는 아래 4가지 조건을 만족해야합니다.

* 길이가 2를 만족하면서
* 각 자리의 문자는 로마자 대문자를 만족하고
* 특수 문자 또는 공백이 없는 문자.
* 아래의 정규표현식을 만족합니다

<세부 카테고리> ::= [A-Z]{2}

**생성규칙:** 길이가 2인 문자열로 생성되며, 상품을 등록할 때 아무런 문제가 없다면 같은 카테고리 코드를 가진 상품 중 프로그램에 등록된 순서로 각각 AA부터 ZZ까지 순서대로 추가. 카테고리 코드가 다르다면 중복생성이 가능하며, 생성할 수 있는 모든 경우를 사용하면 더 이상 생성할 수 없습니다.

**동치 비교 :** 두 세부 카테고리 코드 간의 동치 비교를 할 때에는, 두 문자열 전체가 완전히 동일한 로마자 대문자로 이루어진 문자열인 경우 동일한 것으로 간주합니다.

**검색 :** 입력한 검색어를 세부 카테고리 코드중에서 찾을때에는:

* 검색어와 세부 카테고리 코드 모두 로마자 대문자로 치환하고
* 검색어가 세부 카테고리 코드의 문자열과 완전히 같다면, 매치된 것으로 간주합니다.

**5.7 가격**

**문법 형식 :** 문법적으로 올바른 가격 형식은 아래 조건에 모두 부합하는 문자열입니다:

* 모든 문자는 아라비아 숫자여야 합니다.
* 길이는 1이상 8이하로 즉, 1억원 미만의 단위만 사용 가능합니다.
* 선행 0은 허용하지 않습니다.
* 정규표현식으로 표현하면 다음과 같습니다.

<가격> ::= ( [1-9][0-9]{0,7} | 0 )

**5.8 날짜**

이 기획서에서는 “날짜”란 포스기 최초 입력값과 상품 등록시 입력값을 의미합니다. 2000년 1월 1일 부터 2099년 12월 31일 까지의 날짜만 입력 가능합니다.

**문법 형식( 포스기 최초 입력값, 상품 등록시 입력값) :** 문법적으로 올바른 날짜 두개는 아래 조건에 모두 부합하는 문자열입니다.

* 길이가 8를 만족합니다.
* 모든 문자는 아라비아 숫자여야 합니다.
* 선행 0은 허용하지 않습니다.
* 아래의 정규표현식을 만족해야 합니다.

<날짜> ::= 20[0-9]{2}( 02( 0[1-9] | 1[0-9] | 2[0-8] ) | ( 0( 4 | 6 | 9 )| 11 )( 0[1-9] | [12][0-9] | 30 ) | ( 0( 1 | 3 | 5 | 7 | 8 ) | 1( 0 | 2 ))( 0[1-9] | [12][0-9] | 3[01]))

**5.9 유통기한 설정 값**

유통기한 설정 값은 상품 등록 이후 해당 상품의 유통기한을 설정해주기 위한 값입니다.

ex) 과자 유통기한 설정 값: 30일, 상품 등록일: 20201010 -> 유통기한: 20201109

**문법 형식 :** 문법적으로 올바른 유통기한 설정 값 형식은 아래 조건에 모두 부합하는 문자열입니다:

* 모든 문자는 아라비아 숫자여야 합니다.
* 첫 문자는 1 이상의 숫자여야 합니다.
* 한 자리 부터 세 자리까지 입력가능 합니다.
* 선행 0은 허용하지 않습니다.
* 아래의 정규표현식을 만족해야 합니다.

<유통기한 설정 값> ::= [1-9][0-9]{0,2}

**5.10 수량**

**문법 형식(재고 관리) :** 문법적으로 올바른 수량은 아래 조건에 모두 부합하는 문자열입니다:

* 모든 문자는 아라비아 숫자여야 합니다.
* 첫 문자는 1 이상의 숫자여야 합니다.
* 한 자리 부터 세 자리, 1부터 최대 676까지 입력가능 합니다.
* 선행 0은 허용하지 않습니다.
* 아래의 정규표현식을 만족해야 합니다.

<수량> ::= [1-9][0-9]{0,1} | [1-5][0-9]{2} | 6( [0-6][0-9] | 7[0-6] )

**문법 형식(지폐 추가) :** 문법적으로 올바른 수량은 아래 조건에 모두 부합하는 문자열입니다:

* 모든 문자는 아라비아 숫자여야 합니다.
* 첫 문자는 1이상의 숫자여야 합니다.
* 한 자리 부터 두 자리, 1부터 최대 99까지 입력가능 합니다.
* 선행 0은 허용하지 않습니다.
* 아래의 정규표현식을 만족해야 합니다.

<수량> ::= [1-9][0-9]{0,1}

**6. 데이터 파일**

이 프로그램은 결제와 환불, 재고관리 프로그램 실행 중 사용자로부터 키 입력으로 받아들이고 이 정보들을 프로그램 스스로 내부 기능을 통해 데이터 파일에 저장 및 편집 합니다.

사용자가 외부 편집기를 이용해서 데이터 파일을 직접 생성/편집/저장하는 방식은 지원하지 않습니다.

**7. 주 프롬프트**

프로그램을 정상적으로 실행하면 화면에 다음과 같은 프롬프트가 출력된 상태입니다:

종료하시려면 “종료”를 입력해주세요.

날짜입력 :

**문법 형식** : 입력한 명령어는 길이가 2 또는 8인 문자열이어야 합니다.

**의미 규칙** : 의미규칙은 다음 두 가지 중 하나를 준수해야 합니다.

1.입력한 명령어는 길이가 8이고 데이터요소 ‘날짜’ 형식을 준수해야 합니다.

2.입력한 명령어는 반드시 문자열 “종료”와 일치해야 합니다.

**비정상 결과** : 입력한 명령어가 문법 형식에 어긋난다면 길이가 작거나 크다고 프롬프트에 출력한 뒤 다시 입력을 받습니다. 그리고, 문볍 형식을 만족하더라도, 입력한 명령어가 두 가지 의미 규칙 중 어느 것도 일치하지 않으면 잘못된 입력이라고 프롬프트에 출력합니다. 그 후 다시 입력을 받습니다.

**정상 결과** : 두 가지 의미규칙에 따라, 정상 결과는 두 가지로 나뉩니다.

“종료”를 입력했을 경우 이 프롬프트가 종료되며 본 프로그램이 종료됩니다.

날짜를 입력했을 경우 데이터파일에 저장되어있는 최근 날짜와 비교합니다. 만약 입력한 날짜가 데이터 파일에 저장되어있는 최근 날짜보다 이른 날짜라면 이미 지난 날짜라고 프롬프트에 출력합니다. 그 후 다시 입력을 받습니다. 만약 같은 날짜이거나 데이터 파일에 저장되어있는 최근 날짜 이후의 날짜라면 정상적으로 “메인메뉴”로 넘어갑니다.

날짜를 20XXYYZZ라고 정상적으로 입력했다면 다음과 같은 프롬프트가 출력된 상태입니다.(XX: 년, YY: 월, ZZ: 일):

20XX\_YY\_ZZ

1.결제하기

2.환불하기

3.재고관리

4.현금관리

5.매출확인

6.종료

해당 프롬프트는 “메뉴 선택 방식”으로 사용하는 프로그램의 “메인 메뉴”에 해당합니다.

**문법 형식 :** 입력한 명령어는 길이가 1인 정수이어야 합니다. 명령어로서 적법한 단어들은 아래 <표 1>에 나오는 6개입니다.

|  |  |
| --- | --- |
| 명령어 | 설명 |
| 1 | 상품코드를 입력하여 결제합니다. |
| 2 | 결제코드와 상품코드를 입력하여 상품을 환불합니다. |
| 3 | 재고를 관리합니다. |
| 4 | 현금(지폐)를 충전합니다. |
| 5 | 일 매출을 확인합니다. |
| 6 | “메인메뉴” 프롬프트를 종료합니다. |

<표 1>”메인메뉴” 프롬프트의 명령어들

**의미 규칙 :** 입력한 키가 정수 ‘1’부터 ‘6’, 즉 정수 6개 중 하나와 완전히 일치해야 합니다.

**비정상 결과 :** 만일 입력한 단어가 위 <표 1>의 명령어들과 다를 경우 오류 메세지를 출력합니다. 그리고 다시 “메인 메뉴”에서 키를 입력받습니다.

**정상 결과 :**

1.입력한 명령어가 ‘1’일 경우 7.1.1 부 프롬프트 1 상품코드 입력으로 넘어갑니다.

2.입력한 명령어가 ‘2’일 경우 7.2.1 부 프롬프트 1 : 결제 코드 입력으로 넘어갑니다.

3.입력한 명령어가 ‘3’일 경우 7.3.1 부 프롬프트 : 재고 관리 세부 명령 선택으로 넘어갑니다.

4.입력한 명령어가 ‘4’일 경우 7.4.1 부 프롬프트 1 : 현금 충전 세부 명령 선택으로 넘어갑니다.

5.입력한 명령어가 ‘5’일 경우 7.5.1 부 프롬프트 : 매출 조회로 넘어갑니다.

6.입력한 명령어가 ‘6’일 경우 “종료 명령”으로 넘어갑니다.

**7.1 결제 명령 : ‘1’**

**7.1.1 부 프롬프트 1 : 상품코드 입력**

해당 프롬프트는 결제처리를 해주는 프롬프트입니다. 정상적인 상품코드를 입력할 때마다 결제 리스트가 프롬프트에 출력됩니다.

**문법 형식 :** 입력한 명령어는 문자열의 길이가 2이거나 6이어야 합니다.

**의미 규칙 :** 의미 규칙은 다음 두 가지 중 하나를 따라야 합니다.

1.입력한 명령어는 상품코드 형식을 준수해야 합니다.

2.입력한 명령어는 문자열 “완료”와 반드시 일치해야 합니다.

**비정상 결과 :** 문법 형식이나 의미 규칙에 맞지 않는 입력에 대해 잘못된 입력이라는 메세지를 프롬프트에 출력하고 다시 문법형식을 준수한 명령어를 입력받습니다.

두 가지 형식의 입력방식에 따라 정상 결과는 두 가지로 나뉩니다.

**상품코드 정상 입력 결과 :** 상품코드가 데이터파일에 등록된 코드인지 판별합니다. 만약 코드가 존재한다면 해당 코드의 상품명이 정상적으로 프롬프트에 업데이트되고 결제 대상으로 지정됩니다. 코드가 존재하지 않는다면 잘못된 상품코드라는 메세지를 프롬프트에 출력합니다. 그 후 다시 명령어를 입력받습니다. 만약 현재 프롬프트에서 이미 입력한, 즉 결제 리스트에 이미 존재하는 상품코드를 입력했을 경우, 수량을 늘려줍니다. 그 후 다시 명령어를 입력합니다.

**“완료” 정상 입력 결과 :** 지금까지 입력한 상품코드들을 결제하기 위해 다음 7.1.2 부 프롬프트 2로 넘어갑니다.

**7.1.2 부 프롬프트 2 : 결제수단 선택 명령**

다음과 같이 프롬프트에 출력이 됩니다.

결제수단을 선택해주세요(1.현금, 2. 카드):

**문법 형식 :** 해당 명령어는 길이가 1인 정수입니다.

**의미 규칙 :** 입력한 명령어가 정수 ‘1’ 혹은 ‘2’와 완전히 일치해야 합니다.

**비정상결과 :** 의미규칙을 준수하지 않은 입력을 했다면, 잘못된 입력이라고 메세지를 프롬프트에 출력하고 다시 입력을 받습니다.

**정상결과 :** 입력한 명령어가 의미 규칙을 준수했다면 7.1.3 부 프롬프트 3으로 넘어갑니다.

**7.1.3 부 프롬프트 3 : 결제할지 묻기**

해당 프롬프트는 7.1.1 부 프롬프트 1, 7.1.2 부 프롬프트 2에서 입력한 상품코드들, 결제수단정보을 출력해주고 이를 결제할지 묻는 프롬프트입니다.

**문법 형식 :** 입력한 명령어는 길이가 1인 문자열이어야 합니다.

**의미 규칙 :** 입력한 명령어는 Y/N 형식을 준수해야 합니다.

**비정상 결과 :** 문법 형식에 어긋날 경우, 잘못된 입력이라는 메세지를 프롬프트에 출력합니다. 그후 다시 입력을 받습니다.

**정상 결과 :** 문법 형식을 준수한 두 가지 입력방식에 따라 정상 결과는 다음과 같습니다.

**입력한 명령어가 ‘Y’일 경우 :** 7.1.2 부프롬프트 2에서 ‘1’을 입력했다면 7.1.4 부 프롬프트 4로 넘어갑니다. 만약 ‘2’를 입력했다면 데이터파일에 결제코드와 관련된 정보들을 저장합니다. 결제가 완료되고 결제코드가 출력됩니다, 그후 “메인메뉴”로 넘어갑니다.

**입력한 명령어가 ‘N’일 경우 :** 결제 명령을 종료하고 “메인 메뉴”로 돌아갑니다.

이외의 경우, 잘못된 입력이라고 프롬프트에 출력합니다. 그후 다시 입력을 받습니다.

**7.1.4 부 프롬프트 4 : 금액 입력(현금)**

해당 명령은 현금 결제를 해야할 경우에만 실행됩니다. 실행 시, 결제해야할 모든 상품의 가격을 합친 금액을 출력합니다.

**문법 형식 :** 0 또는 1자리의 정수부터 8자리의 양의 정수입니다. 선행 0은 허용하지 않습니다.

**의미 규칙 :** 문법 형식을 준수하되, 프롬프트에 출력된 금액 이상의 값을 입력해야 합니다.

**비정상 결과 :** 명령어가 문법 형식에 위배된다면 잘못된 메세지라고 프롬프트에 출력된 뒤 다시 입력을 받습니다. 혹은 문법 형식을 준수하되 의미 규칙을 위반한다면 금액이 부족하다는 메세지를 프롬프트에 출력합니다. 그 후 다시 입력받습니다.

**정상 결과 :**

1. 입력한 금액에서 지불해야할 금액을 뺀 거스름돈을 계산한 뒤 이를 출력합니다.

2.데이터파일에 저장되어있는 지폐, 동전 개수를 거스름돈을 계산하여 차감합니다.

2.1 이 때 포스기의 현금보유잔량이 거스름돈보다 부족할 경우 충전을 위해 7.4 부 프롬프트 “충전 명령”으로 넘어갑니다.

2.2 충전을 정상적으로 완료한 뒤 다시 7.1.4 부프롬프트 4로 돌아온 뒤 기존 결제정보를 토대로 결제를 진행합니다.

3.결제가 완료되고 결제코드가 출력됩니다, 그후 “메인메뉴”로 넘어갑니다.

4.데이터파일에 결제코드와 관련된 정보들을 저장합니다.

**7.2 환불 명령 : ‘2’**

**7.2.1 부 프롬프트 1 : 결제 코드 입력**

해당 프롬프트는 환불처리를 해주는 프롬프트입니다.

**문법 형식 :** 입력한 명령어는 문자열의 길이가 9인 문자열이어야 합니다.

**의미 규칙 :** 입력한 명령어는 결제코드 형식을 준수해야 합니다.

**비정상 결과 :** 입력한 명령어가 문법형식에 어긋나거나 의미 규칙을 지키지 않았다면잘못된 입력이라는 메세지를 프롬프트에 출력하고 다시 입력받습니다.

**정상 결과 :** 결제코드가 존재하는지 판단을 합니다. 만약 결제코드가 존재한다면 환불이 가능한지 판단합니다. 환불이 가능하다면 환불 가능 리스트를 출력하고, 7.2.2 부 프롬프트 2로 넘어갑니다. 만약 환불이 불가능하다면 프롬프트에 환불이 불가능하다고 출력 후 프롬프트 종료 및 “메인 메뉴”로 돌아갑니다. 결제코드가 존재하지 않으면 결제코드가 존재하지 않다고 프롬프트에 출력하고 다시 입력받습니다.

**7.2.2 부 프롬프트 2 : 상품코드 입력**

데이터파일에 존재하는 결제코드를 입력할 시 실행됩니다. 해당 프롬프트에 환불이 가능한 상품(코드)목록들을 출력합니다.

**문법 형식 :** 입력한 명령어는 문자열의 길이가 2 혹은 6인 문자열이어야 합니다.

**의미 규칙 :** 의미 규칙은 다음 두 가지 중 하나를 따릅니다.

1.입력한 명령어는 상품코드 형식을 준수해야합니다.

2.입력한 명령어는 문자열 “완료”와 반드시 일치해야 합니다.

**비정상 결과 :** 잘못된 입력이라는 메세지를 프롬프트에 출력하고 다시 문법형식을 준수한 명령어를 입력받습니다.

**정상 결과 :**

두 가지 형식의 입력방식에 따라 정상 결과는 두 가지로 나뉩니다.

**상품코드 정상 입력 결과 :** 상품코드가 결제코드에 등록된 코드인지 판별합니다. 만약 코드가 존재한다면 7.2.3 부 프롬프트 3으로 넘어갑니다. 코드가 존재하지 않는다면 잘못된 상품코드라는 메세지를 프롬프트에 출력합니다. 그 후 다시 문법규칙을 준수한 명령어를 입력받습니다.

**“완료” 정상 입력 결과 :** “메인 메뉴”로 돌아갑니다.

**7.2.3 부 프롬프트 3 : 환불여부 묻기**

부 프롬프트 1에서 입력받은 상품에 대해 환불을 할건지에 대해 질문을 해당 프롬프트에 출력합니다.

**문법 형식 :** 입력한 명령어는 길이가 1인 문자열이어야 합니다.

**의미 규칙 :** 입력한 명령어는 Y/N 형식을 준수해야 합니다.

**비정상 결과 :** 문법 형식에 어긋날 경우, 잘못된 입력이라는 메세지를 프롬프트에 출력합니다. 그후 다시 입력을 받습니다.

**정상 결과 :** 문법 형식을 준수한 두 가지 입력방식에 따라 정상 결과는 다음과 같습니다.

**입력한 명령어가 ‘Y’일 경우 :** 환불 명령 프롬프트에서 입력한 결제코드를 토대로 결제수단과 결제기록을 가져옵니다. 그후 결제기록에서 해당 상품코드를 제거합니다. 결제수단이 현금일경우

1.포스기가 보유하고 있는 현금잔량에서 상품가격을 뺍니다.

1.1. 단, 현금잔량이 상품가격보다 적다면 7.4 충전 명령으로 넘어가서 충전을 진행합니다.

1.2. 충전을 정상적으로 했다면 다시 7.2.3 부 프롬프트 3으로 돌아와서 기존 환불정보를 토대로 환불을 진행합니다.

2.환불이 정상적으로 처리되었다고 프롬프트에 출력합니다. 7.2.2 부 프롬프트 2로 넘어갑니다.

**입력한 명령어가 ‘N’일 경우 :** 환불 명령을 종료하고 7.2.2 부 프롬프트 2로 돌아갑니다**.**

이외의 경우, 잘못된 입력이라고 프롬프트에 출력합니다. 그 후 다시 입력을 받습니다.

**7.3 재고관리 명령 : ‘3’**

**7.3.1 부 프롬프트 1 : 재고 관리 세부 명령 선택**

해당 프롬프트는 재고를 관리하는 프롬프트입니다. 데이터파일에 등록되어있는 모든 재고와 다음과 같은 출력문이 프롬프트에 출력됩니다.

1. 상품등록
2. 재고추가
3. 폐기등록
4. 종료

해당 프롬프트는 “메뉴 선택 방식”으로 사용합니다

문법 형식 : 입력한 명령어는 길이가 1인 정수이어야 합니다. 명령어로서 적법한 단어들은 아래 <표 2>에 나오는 4개입니다.

|  |  |
| --- | --- |
| 명령어 | 설명 |
| 1 | 상픔명과 카테고리명을 입력하여 정보를 등록합니다. |
| 2 | 상품정보를 토대로 재고를 추가합니다. |
| 3 | 폐기처리할 상품을 등록합니다. |
| 4 | 재고관리 명령을 종료합니다. |

**<표 2>**

**비정상 결과 :** 만일 입력한 단어가 위 <표 2>의 명령어들과 다를 경우 오류 메세지를 출력합니다. 그리고 다시 재고 관리 세부 명령 선택 창에서 키를 입력받습니다.

**정상 결과 :**

1.입력한 명령어가 ‘1’일 경우 7.3.2 부 프롬프트 2.1 상품명 입력으로 넘어갑니다.

2.입력한 명령어가 ‘2’일 경우 7.3.7 부 프롬프트 3.1 : 상품 재고 추가로 넘어갑니다.

3.입력한 명령어가 ‘3’일 경우 7.3.9 부 프롬프트 4.1 : 폐기할 상품코드 입력으로 넘어갑니다.

4.입력한 명령어가 ‘4’일 경우 7.3.11 부 프롬프트 5 : 종료로 넘어갑니다.

**7.3.2 부 프롬프트 2.1 : 상품명 입력**

해당 부 프롬프트 2.1은 상품정보를 입력하여 등록하는 프롬프트입니다.

**문법 형식 :** 입력한 명령어는 길이가 10 이하인 문자열이어야 합니다.

**의미 규칙 :** 입력한 명령어는 5.3 ‘상품명’ 형식을 준수해야 합니다.

**비정상 결과 :** 한글이 아닌 다른 단어를 입력했을 경우, 잘못된 입력이라는 메세지를 프롬프트에 출력합니다.

**정상 결과 :** 입력된 상품명이 존재하는지 데이터파일에서 검색합니다. 존재한다면 이미 존재하는 상품정보라고 프롬프트에 출력하고 7.3.1 부 프롬프트 : 재고 관리 세부 명령 선택으로 돌아갑니다. 존재하지 않는 경우 7.3.3 부 프롬프트 2.2로 넘어갑니다.

**7.3.3 부 프롬프트 2.2 : 카테고리명 입력**

**문법 형식 :** 입력한 명령어는 길이가 10이하인 문자열여야 합니다..

**의미 규칙 :** 5.4 **‘**카테고리명’ 형식을 준수해야 합니다.

**비정상 결과 :** 입력한 명령어의 길이가 11 이상이면 길이가 길다고 프롬프트에 출력합니다. 의미 규칙이 어긋나면 잘못된 입력이라고 프롬프트에 출력합니다.

.

**정상 결과 :** 입력한 카테고리명이 데이터파일에 존재하는지 검색합니다.

존재하지 않는다면 카테고리 코드와 세부 카테고리 코드를 생성하고 7.3.4 부 프롬프트 2.3으로 넘어갑니다. 존재한다면 기존 카테고리 코드를 가져오고 세부 카테고리 코드를 생성하고 7.3.4 부 프롬프트 2.3으로 넘어갑니다.

**7.3.4 부 프롬프트 2.3 : 유통기한 입력**

**문법 형식 :** 입력한 명령어는 길이가 3 이하인 문자열이어야 합니다.

**의미 규칙 :** 5.9 ‘유통기한 설정 값’ 형식을 준수해야합니다.

**비정상 결과 :** 입력한 명령어의 문자열이 4 이상일 경우 길이가 길다고 프롬프트에 출력합니다. 의미 규칙에 어긋난다면 잘못된 입력이라고 프롬프트에 출력합니다.

**정상 결과 :** 7.3.5부 프롬프트 2.4로 넘어갑니다.

**7.3.5 부 프롬프트 2.4 : 가격입력**

**문법 형식 :** 입력한 명령어는 길이가 1이상 8이하인 0 이상의 정수여야 하고, 선행 0은 허용되지 않습니다.

**의미 규칙 :** 이 과정은 문법 형식 외에 더 준수해야 할 의미 규칙이 없습니다.

**비정상 결과 :** 잘못된 입력이라는 메세지를 프롬프트에 출력하고, 다시 입력을 받습니다.

**정상 결과 :** 7.3.6부 프롬프트 2.5로 넘어갑니다.

**7.3.6 부 프롬프트 2.5 : 상품정보를 추가할지 묻기**

해당 프롬프트는 7.3.2 부 프롬프트 2.1에서 7.3.5 부 프롬프트 2.4까지 입력한 정보를 데이터파일에 저장할 지 묻는 프롬프트입니다. 상품명, 카테고리명, 카테고리코드, 세부카테고리, 유통기한 설정값, 가격을 프롬프트에 출력하고 이를 저장할지 묻습니다.

**문법 형식 :** 입력한 명령어는 길이가 1인 문자열이어야 합니다.

**의미 규칙 :** 입력한 문자열은 Y/N 형식을 준수해야 합니다.

**비정상 결과 :** 문법 형식에 어긋날 경우, 잘못된 입력이라는 메세지를 프롬프트에 출력합니다. 그후 다시 입력을 받습니다.

**정상 결과 :**

**입력한 명령어가 ‘Y’일 경우:** 지금까지 입력 또는 수정 또는 생성한 상품명, 카테고리명, 카테고리코드, 세부카테고리, 유통기한 설정값, 가격을 데이터파일에 저장합니다. 그리고 등록되었다고 프롬프트에 출력합니다.

그 후 7.3.1 부 프롬프트 : 재고 관리 세부 명령 선택으로 돌아갑니다.

**입력한 명령어가 ‘N’일 경우:** 재고 관리 세부 명령 선택창으로 돌아갑니다.

이외의 경우, 잘못된 입력이라고 프롬프트에 출력합니다. 그후 다시 입력을 받습니다.

**7.3.7 부 프롬프트 3.1 : 상품 재고 추가**

상품명을 입력받아 해당 상품정보를 토대로 재고를 추가하는 프롬프트입니다.

**문법 형식 :** 입력한 명령어는 길이가 10 이하인 문자열이어야 합니다.

**의미 규칙 :** 입력한 명령어는 상품명 형식을 준수해야 합니다.

**비정상 결과 :** 한글이 아닌 다른 단어를 입력했을 경우, 잘못된 입력이라는 메세지를 프롬프트에 출력합니다.

**정상 결과 :** 입력한 상품명이 데이터파일에 존재하는지 검색합니다.

존재하지 않는다면 없는 상품명이라고 프롬프트에 출력하고 7.3.1 부프롬프트 1 : 재고 관리 세부 명령 선택으로 되돌아갑니다. 존재한다면 현재 해당 상품의 수량이 676개인지 확인합니다. 676개가 아니라면 7.3.8 부 프롬프트 3.2로 넘어갑니다. 만일, 676개라면 추가 불가 메세지를 출력하고, 7.3.1 부프롬프트 1 : 재고 관리 세부 명령 선택으로 되돌아갑니다.

**7.3.8 부 프롬프트 3.2 : 상품 재고 수량 입력**

부 프롬프트 2.1에서 입력한 상품의 재고를 입력하는 프롬프트입니다.

**문법 형식 :** 입력한 명령어는 길이가 3이하인 문자열이어야 합니다.

**의미 규칙 :** 입력한 명령어는 1부터 676까지 음이 아닌 정수이어야 합니다.

**비정상 결과 :** 입력한 명령어의 길이가 4이상이면 길이가 길다고 프롬프트에 출력됩니다. 의미 규칙에 어긋난다면 잘못된 입력이라고 프롬프트에 출력합니다.

**정상 결과 :** 입력한 수량만큼 7.3.7 부 프롬프트 3.1에서 입력한 상품명이 지닌 정보들을 토대로 재고를 추가합니다. 단, 상품코드 생성 규칙에 따라 같은 상품명은 최대 676개까지 저장할 수 있으므로 입력한 수량이 이를 초과한다면 최대 676개까지 데이터파일에 저장합니다. 그 후 재고가 추가되었다고 출력하고 7.3.1 부프롬프트 1 : 재고 관리 세부 명령 선택으로 돌아갑니다.

**7.3.9 부 프롬프트 4.1 : 폐기할 상품코드 입력**

해당 프롬프트에 진입하면 유통기한이 지난 상품코드들이 출력됩니다. 출력된 상품코드 중에서 사용자가 선택하여 입력해야 합니다.

**문법 형식 :** 입력한 명령어는 길이가 2 혹은 6인 문자열이어야 합니다.

**의미 규칙 :** 상품코드 형식을 준수한 문자열이어야 합니다.

**비정상 결과 :** 입력한 명령어와 프롬프트에 출력된 상품코드들 중 어느 것도 일치하지 않다면 잘못된 입력이라고 프롬프트에 출력합니다. 그후 다시 상품코드를 입력받습니다.

**정상 결과 :** 입력한 명령어가 출력된 상품코드들 중 하나와 일치한다면 7.3.10 부 프롬프트 4.2로 넘어갑니다.

혹은, 입력한 명령어가 “완료”와 일치한다면 7.3.1 부프롬프트 1 : 재고 관리 세부 명령 선택으로 돌아갑니다.

**7.3.10 부 프롬프트 4.2 : 폐기할지 묻기**

해당 프롬프트에서는 7.3.9 부 프롬프트 4.1에서 입력한 상품(코드)를 폐기할지 묻는 프롬프트입니다. 진입 시 프롬프트에 해당 상품을 폐기할지 묻는 질문을 출력합니다.

**문법 형식 :** 입력한 명령어는 길이가 1인 문자열이어야 합니다.

**의미 규칙 :** 입력한 명령어는 Y/N 형식을 준수해야 합니다.

**비정상 결과 :** 문법 형식에 어긋날 경우, 잘못된 입력이라는 메세지를 프롬프트에 출력합니다. 그후 다시 입력을 받습니다.

**정상 결과 :**

**입력한 명령어가 ‘Y’일 경우 :** 해당 상품코드는 재고에서 폐기처리됩니다. 즉, 데이터파일에서 해당 상품코드를 찾아낸 뒤 이를 삭제합니다. 이후 7.3.9 부 프롬프트 4.1로 돌아갑니다.

**입력한 명령어가 ‘N’일 경우 :** 7.3.9 부 프롬프트 4.1로 돌아갑니다.

이외의 경우, 잘못된 입력이라고 프롬프트에 출력합니다. 그후 다시 입력을 받습니다.

**7.3.11 부 프롬프트 5 : 종료**

해당 프롬프트에서는 재고관리 프롬프트를 종료하는 프롬프트 입니다.

**문법 형식 :** 길이가 1인 문자열이어야 합니다.

**의미 규칙 :** 입력한 문자열이 “4”이어야만 합니다.

**비정상 결과 :** 입력한 문자열이 의미규칙과 다르다면, 오류메세지를 출력하거나, 다른 작업이 수행될 수 있습니다.

**정상 결과 :** 입력한 문자열이 의미규칙과 일치한다면, 메인 메뉴로 다시 돌아갑니다.

**7.4 현금 충전 명령 : ‘4’**

**7.4.1 부 프롬프트 1 : 현금 충전 세부 명령 선택**

해당 프롬프트에 진입 시 데이터파일에 저장되어 있는 각 10원, 50원, 100원, 500원, 1000원, 5000원, 10000원, 50000원의 개수(장수)를 프롬프트에 출력합니다.

그 후, 현금 충전을 위해 다음과 같은 출력문이 출력됩니다.

1.10원

2.50원

3.100원

4.500원

5.1000원

6.5000원

7.10000원

8.50000원

충전할 지폐(동전)을 선택해주세요 :

**문법 형식 :** 입력한 명령어는 길이가 1인 0이 아닌 양의 정수이거나 길이가 2인 문자열이어야 합니다.

**의미 규칙 :** 입력한 정수는 반드시 1,2,3,4,5,6,7 그리고 8 중 하나와 일치하거나 “완료”와 일치해야 합니다.

**비정상 결과 :** 문법 형식을 준수하지 않으면 잘못된 입력이라고 프롬프트에 출력합니다. 첫번째 의미 규칙에 어긋난다면 1~8만 입력하라고 프롬프트에 출력합니다.

**정상 결과 :** 첫번째 의미 규칙에 맞는 명령어를 입력했을 때,

해당 현금 종류 보유수량이 99개가 아니라면 7.4.2 부 프롬프트 2 : 수량입력으로 넘어갑니다.

만일 해당 현금 종류 보유수량이 99개라면, 추가 불가 메시지를 출력하고, 다시 입력을 받습니다.

두번째 의미 규칙에 맞는 명령어를 입력했다면 메인 메뉴로 넘어갑니다.

**7.4.2 부 프롬프트 2 : 수량 입력**

**문법 형식 :** 입력한 명령어는 길이가 2이하인 정수여야 합니다.

**의미 규칙 :** 입력한 명령어는 1부터 99까지 음이 아닌 정수여야 합니다.

**비정상 결과 :** 입력한 명령어의 길이가 3 이상일 경우 길이가 길다고 프롬프트에 출력합니다. 의미 규칙에 어긋난다면 잘못된 입력이라고 프롬프트에 출력합니다.

**정상 결과 :** 만일 해당 현금 종류 보유수량에 추가수량이 합해진 값이 99개 이상이라면, 추가 불가 메시지를 출력하고, 다시 입력을 받습니다.

이외에 의미 규칙에 맞는 명령어를 입력했다면 7.4.3 부 프롬프트 3 : 충전할지 묻기로 넘어갑니다.

**7.4.3 부 프롬프트 3 : 충전할지 묻기**

현금충전 프롬프트와 부 프롬프트 ‘수량입력’에서 입력한 모든 현금종류와 수량을 토대로 현금을 충전할지 묻는 프롬프트입니다. 지금까지 입력한 현금종류와 각 지폐의 수량을 프롬프트에 출력합니다.

**문법 형식 :** 입력한 명령어는 길이가 1인 문자열이어야 합니다.

**의미 규칙 :** 입력한 명령어는 Y/N 형식을 준수해야 합니다.

**비정상 결과 :** 문법 형식에 어긋날 경우, 잘못된 입력이라는 메세지를 프롬프트에 출력합니다. 그후 다시 입력을 받습니다.

**정상 결과 : 입력한 명령어가 ‘Y’일 경우 :** 현금 충전 명령에서 입력한 모든 지폐(동전)의 종류와 입력한 수량을 가져오고 데이터파일에서 현금보유잔량을 가져옵니다. 그후 보유수량과 충전할 현금수량을 더해서 데이터파일에 저장하고 현금이 충전되었다고 프롬프트에 출력합니다. 그 후 7.4.1 부 프롬프트 1로 넘어갑니다.

**7.5 매출 조회 명령 : ‘5’**

**7.5.1 부 프롬프트 : 매출 조회**

해당 프롬프트는 입력한 날짜 하루 동안 일어난 결제기록들을 계산하여 일 매출을 조회하는 프롬프트입니다. 해당 프롬프트는 매출 조회만을 위한 프롬프트이므로, 조회 외에 특별한 기능은 없습니다. 단, 입력은 받되 프롬프트에서 벗어날 수 있는 명령어만 입력받습니다. 추가로 완료를 입력해야 종료할 수 있다고 프롬프트에 출력합니다.

**문법 형식 :** 입력한 명령어는 길이가 2인 문자열입니다.

**의미 규칙 :** 입력한 명령어는 문자열 “완료”와 반드시 일치해야 합니다.

**비정상 결과 :** 입력한 명령어의 길이가 1이하 혹은 3이상이면 길이가 짧거나 길다고 프롬프트에 출력합니다. 의미 규칙에 어긋난다면 완료를 입력해야 한다고 프롬프트에 출력합니다.

**정상 결과 :** 프롬프트를 종료하고 “메인 메뉴”로 넘어갑니다.

**7.6 종료 명령 : ‘6’**

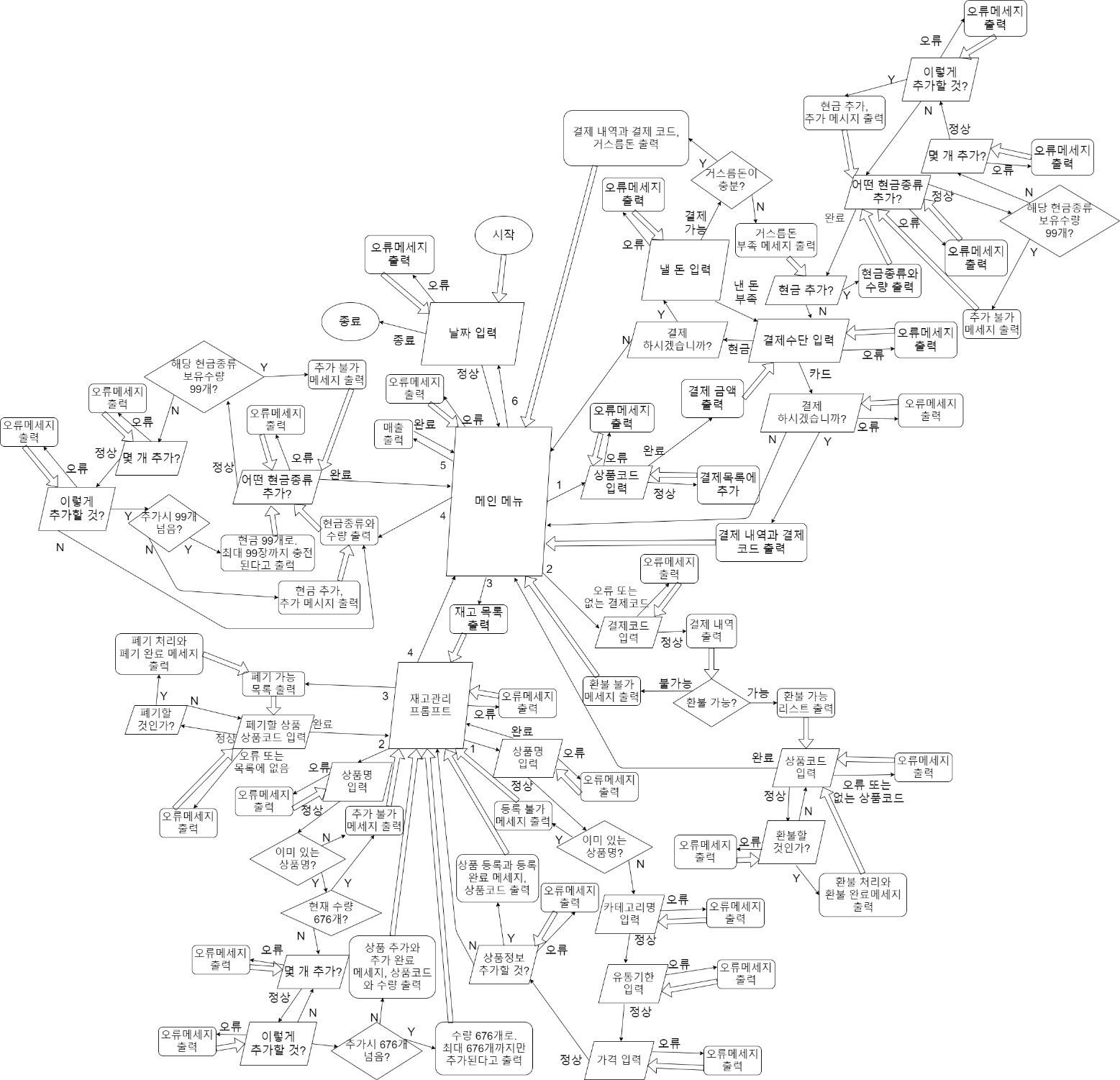
해당 프롬프트는 메인메뉴 프롬프트에서 날짜입력 프롬프트로 이동시켜주는 프롬프트 입니다

**문법 형식 :** 길이가 1인 문자열이어야 합니다.

**의미 규칙 :** 입력한 문자열이 “6”이어야만 합니다

**비정상 결과 :** 만일 인자가 의미규칙과 다르다면, 오류메세지를 출력하거나, 다른 작업이 수행될 수 있습니다.

**정상 결과 :** 인자가 의미규칙과 일치한다면, 모든 부 프롬프트를 종료하고 주 프롬프트 날짜 입력 부분으로 다시 돌아갑니다.



“Figure 1: 사용 흐름도”