

다음과 같은 주의사항을 참고하여 조건에 만족하는 프로그램을 작성한다.

- 패키지명, 클래스명, 메소드명은 반드시 제시한 것으로 작성한다.
 - 클래스 구조는 변경할 수 없다.

상위 사항을 위반하면 오답 및 감점 처리된다.

[문제 1] 다음과 같은 조건을 만족하는 프로그램을 작성하시오.

1. 사용 데이터

```
int [][] array = {  
    { 12, 41, 36 ,56,21 },  
    { 82, 10, 12 ,61,45 },  
    { 14, 16, 18 ,78 ,65 },  
    { 45, 26, 72, 23, 34 },  
};
```

2. 구현 클래스

package 명	class 명	method	설명
com.test01	TestMain01	main	main메소드 안에서 모든 코드 작업 구현

3. 실행 결과

출력 결과는 다음과 같다. (합계와 평균 값은 double로 처리한다.)

합 계 : 767.0

평 균 : 38.35

[문제 2] 다음과 같은 조건을 만족하는 프로그램을 작성하시오.

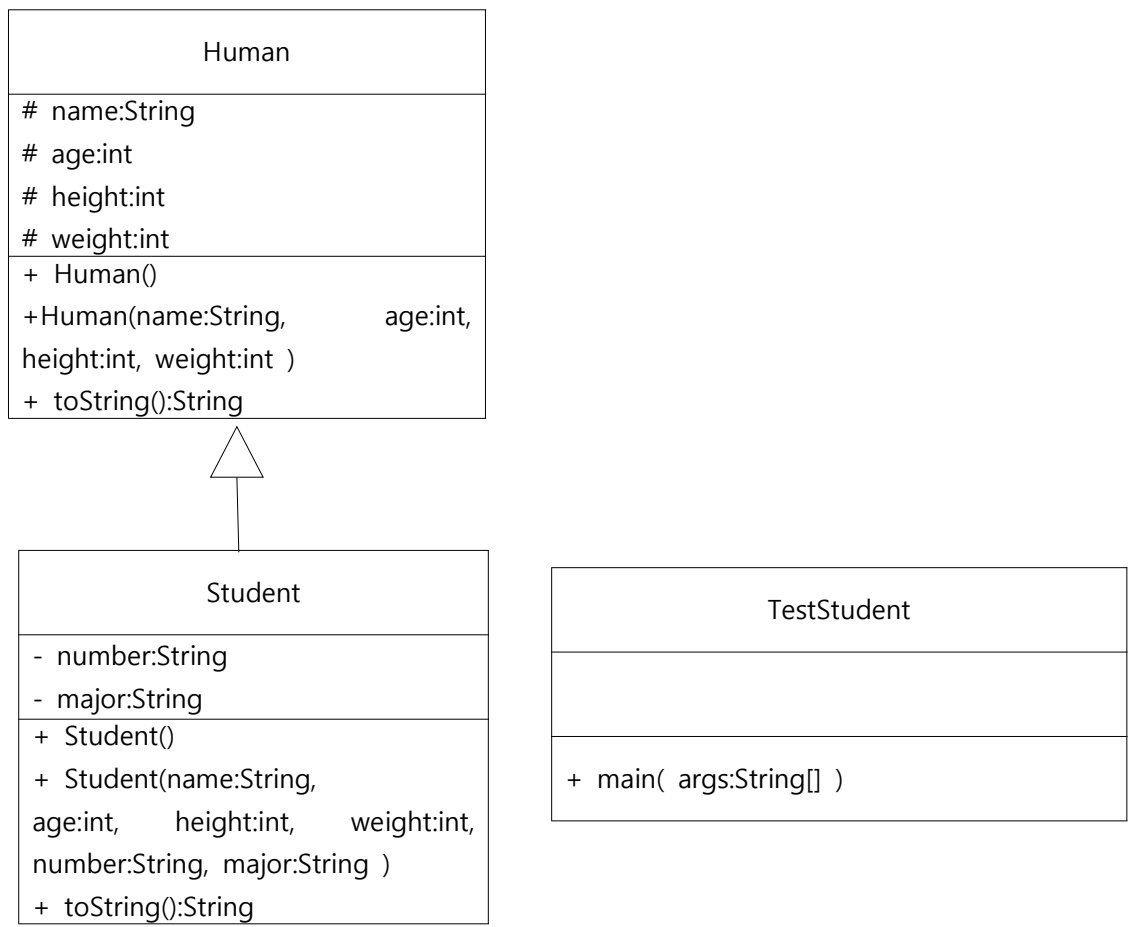
Human 클래스를 상속받은 Student 클래스를 이용하여 프로그래밍 한다.
3개의 Student 객체를 생성하여 List에 저장한 후에 각 객체의 모든 정보를 출력한다.

1. 사용 데이터

아래와 같이 3개의 Student 객체를 생성하여 프로그램을 구현한다.

이름	나이	신장	체중	학번	전공
홍길동	20	177	74	201301	체육
이순신	44	178	77	201302	체육
유관순	18	155	45	201303	컴퓨터

2. 클래스 다이어그램



- * 모든 클래스의 getter, setter 메소드는 직접 구현한다.
- * # 는 protected 접근 지정자이다.

3. 클래스 구조

package 명	class 명	method	설명
com.test02	Human	+Human()	기본 생성자
		+ Human(name:String , age:int,height:int, weight:int)	4개의 인자를 갖는 생성자
		+ toString():String	Human 정보를 리턴한다.
	Student	+ Student()	기본 생성자
		+ Student(name:String, age:int, height:int, weight:int, number:String, major:String)	6개의 인자를 갖는 생성자
		+ toString():String	오버라이딩 메소드
	TestStudent	+ main(args:String[])	main메소드안에서 Student타입의 List 을 선언하여 구현 한다.

4. TestStudent 클래스 구조

Student객체를 담을 수 있는 List를 선언하고 3개의 Student 객체를 생성하여 저장한다.

```
public class TestStudent{
    public static void main( String [] args ){

        List<Student> list = new ArrayList<Student>();
        //Student 객체를 3개 생성하여 List에 저장한다.

        // List에 저장된 Student 객체 정보를 출력한다.

    }
}
```

5. 출력 결과

```
홍길동 20 177 74 201301 체육
이순신 44 178 77 201302 체육
유관순 18 155 45 201303 컴퓨터
```

[문제 3] 다음과 같은 조건을 만족하는 프로그램을 작성하시오.

주어진 String 데이터를 ","로 구분하여 5개의 실수 데이터를 추출하고 합과 평균을 구한다.

(StringTokenizer 사용)

단, String 문자열의 모든 실수 데이터를 List에 저장하고 계산한다.

1. 사용 데이터

```
String str = "1.22,4.12,5.93,8.71,9.34";
```

2. 구현 클래스

package 명	class 명	method	설명
com.test03	TestMain03	main	main메소드 안에서 모든 코드 작업 구현

3. Class 구조

```
public class TestMain03{
    public static void main( String [] args ){

        String str = "1.22,4.12,5.93,8.71,9.34";

        //StringTokenizer 이용하여 List에 저장한다.

        //List에 저장된 데이터의 합과 평균을 구한다.
```

4. 출력결과

합계와 평균은 모두 소수점 4자리에서 반올림하여 소수점 3자리까지 표현한다.

```
합 계: 29.320
평 균: 5.864
```

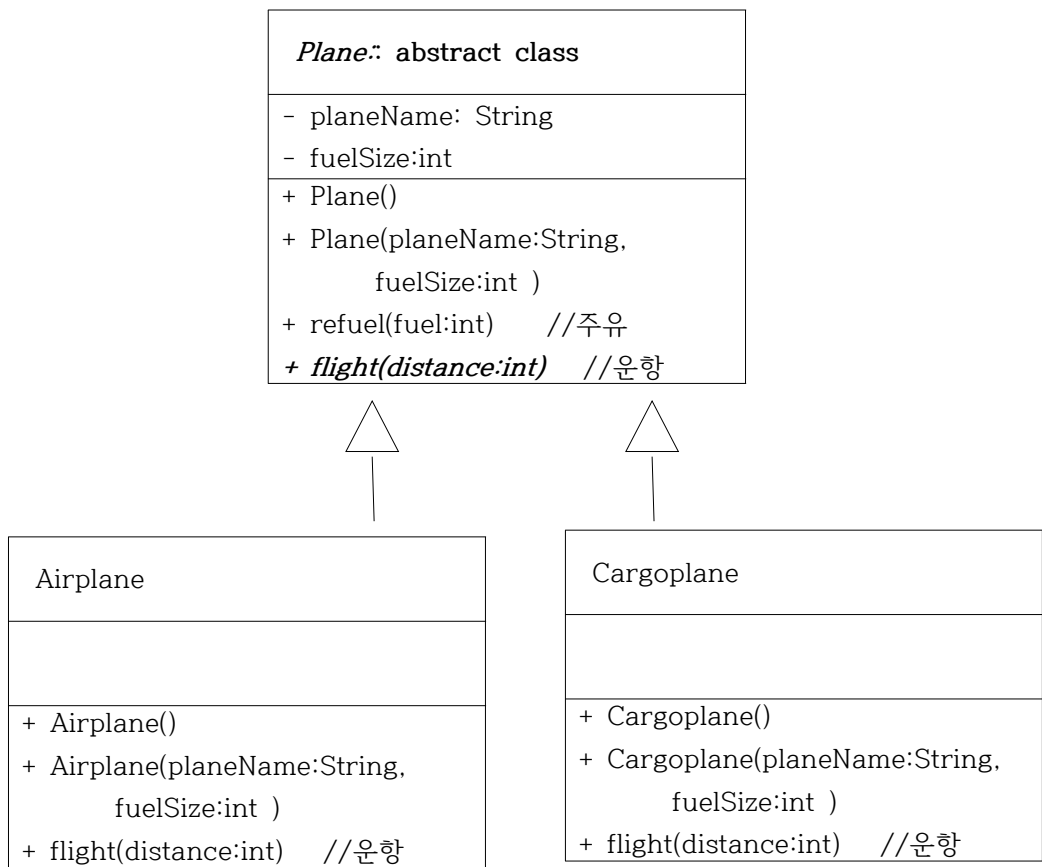
[문제 4] 다음과 같은 조건을 만족하는 프로그램을 작성하시오.

추상클래스에서 상속받은 두 개의 클래스를 구현하여 실행 결과의 예와 같이 출력되도록 프로그램을 작성한다.

1. 사용 데이터

클래스 타입	planeName	fuelSize
Airplane	L777	1000
Cargoplane	C50	1000

2. 클래스 다이어그램



* 모든 클래스의 getter, setter 메소드는 직접 구현한다.

3. 구현 클래스

Package 명	클래스명	메소드	설명
com.test04	Plane	+Plane()	기본 생성자
		+Plane(planeName:String, fuelSize:int)	인자 2개를 갖는 생성자
		+refule(fuel:int)	일정양의 연료 주입, 기존 연료가 증가됨.
		+flight(distance:int)	일정거리만큼 운항, 연료감소. 추상메소드로서 하위클래스에서 반드시 구현
	Airplane	+Airplane()	
		+Airplane(planeName:String, fuel:int)	인자 2개를 갖는 생성자
		+flight(distance:int)	10운항시 연료30감소
	Cargoplane	+Cargoplane()	기본 생성자
		+Cargoplane(planeName:String, fuel:int)	인자 2개를 갖는 생성자
		+flight(distance:int)	10운항시 연료50감소
	PlaneTest	main	main 메소드

4. PlaneTest 클래스 구조

```

public class PlaneTest {

    public static void main(String [] args){

        // Airplane과 Cargoplane 객체생성
        // 생성된 객체의 정보 출력
        // Airplane과 Cargoplane 객체에 100씩 운항
        // 운항후 객체의 변경된 정보 출력
        // Airplane과 Cargoplane 객체에 200씩 주유
        // 주유후 객체의 변경된 정보 출력

    }
}

```

5. 실행 결과

Plane fuelSize

L777 1000
C50 1000
100 운항
Plane fuelSize

L777 700
C50 500
200 주유
Plane fuelSize

L777 900
C50 700

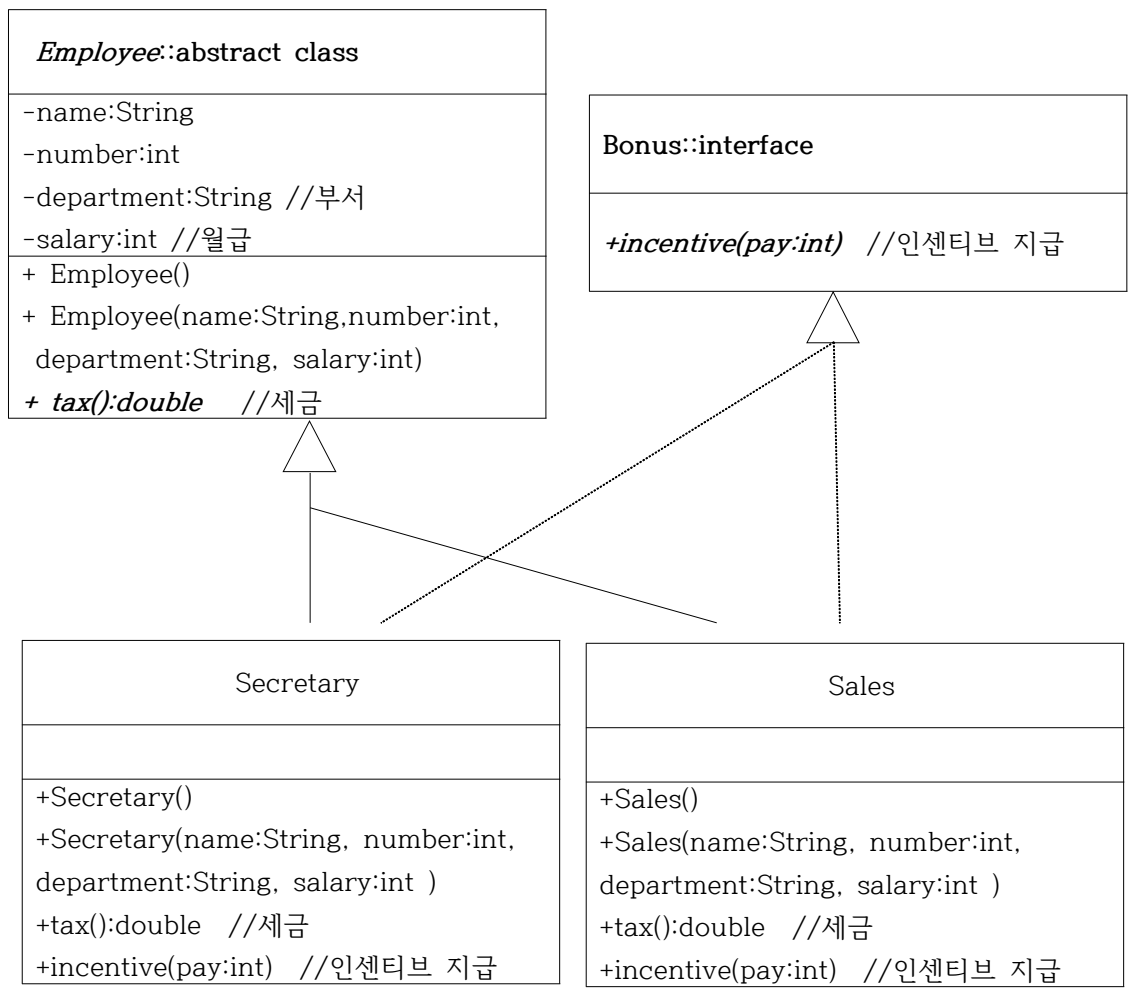
[문제 5] 다음과 같은 조건을 만족하는 프로그램을 작성하시오.

추상 클래스 및 인터페이스를 사용한 두 개의 클래스를 구현하여 실행 결과의 예와 같이 출력되도록 프로그램을 작성한다.

1. 사용 데이터

name	number	department	salary
홍길동	1	Secretary	800
이순신	2	Sales	1200

2. 클래스 다이어그램



3. 구현 클래스

package 명	클래스명	메소드 및 생성자	설명
com.test05	Employee	+Employee()	기본 생성자
		+Employee(...)	4개의 변수를 인자로 갖는 생성자
		+tax():double	세금을 리턴
	Bonus	+incentive(pay:int)	인센티브 지급
	Secretary	+Secretary()	기본 생성자
		+Secretary(...)	4개의 변수를 인자로 갖는 생성자
		+incentive(pay:int)	인센티브 지급. pay에 80%가 기존 salary에 더해진다.
		+tax():double	세금 리턴. salary에 10% 징수
	Sales	+Sales()	기본 생성자
		+Sales(...)	4개의 변수를 인자로 갖는 생성자
		+incentive(pay:int)	인센티브 지급. pay에 120%가 기존 salary에 더해진다.
		+tax():double	세금 리턴. salary에 13% 징수
	Company	+main(args:String[])	main 메소드

* class 명과 메소드명은 변경하지 않는다.

* 모든 클래스변수의 getter, setter 메서드는 직접 구현한다.

4. Company 클래스 구조

<pre> public class Company{ public static void main(String [] args){ HashMap<Integer, Employee> map = new HashMap<>(); // 1번의 데이터를 기반으로 객체를 생성하여 HashMap에 저장한다. // HashMap에 저장시 키 값은 각 객체의 Number로 한다. // 모든 객체의 기본 정보를 출력한다. (for문 이용 , keySet() 이용) // 모든 객체의 인센티브 100씩 지급한다. // 모든 객체의 정보와 세금을 출력한다. (for문 이용) </pre>

5. 실행 결과

name	department	salary
홍길동	Secretary	800
이순신	Sales	1200

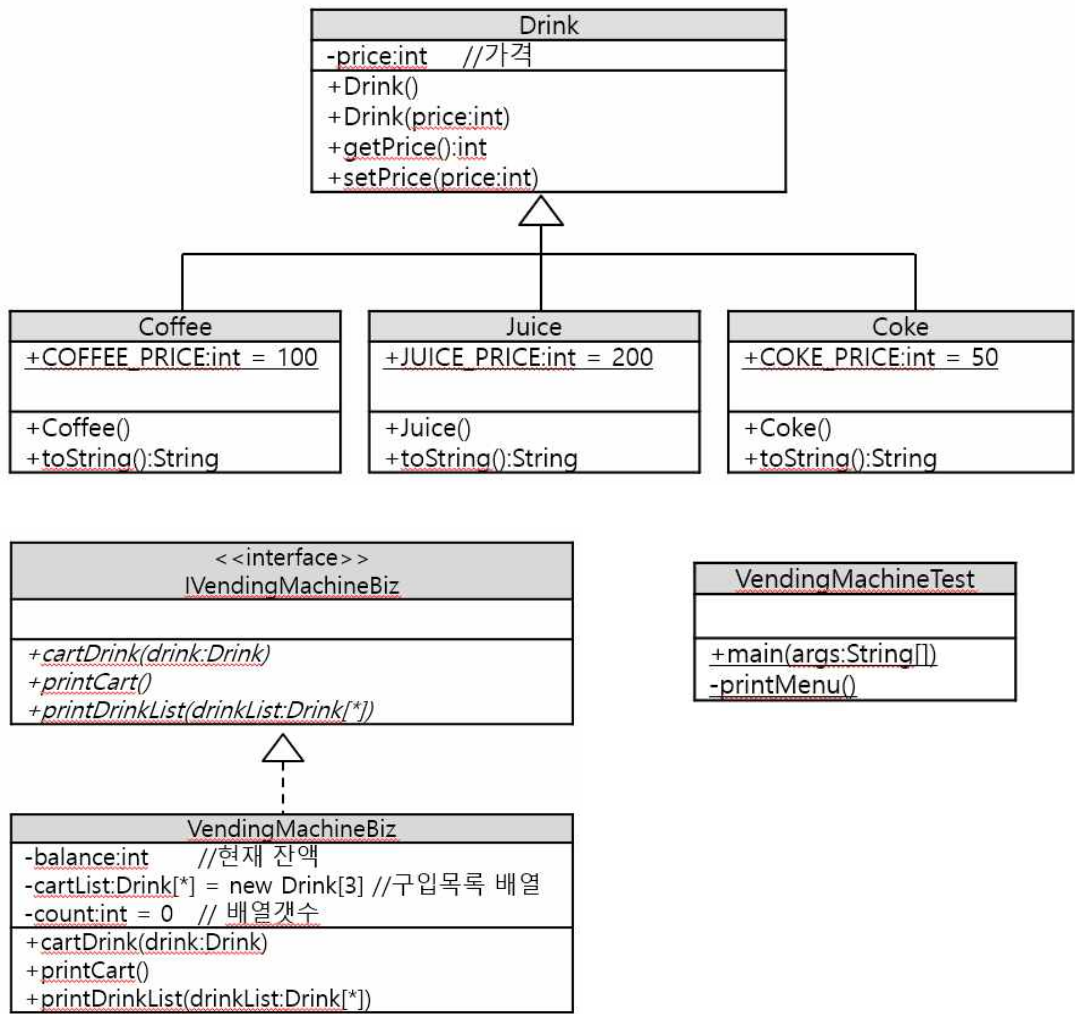
인센티브100지급

name	department	salary	tax
홍길동	Secretary	880	88.00
이순신	Sales	1320	171.60

[문제 6] 다음과 같은 조건을 만족하는 자판기 시스템 프로그램을 작성하시오.

음료수 자판기내에는 커피,쥬스,코크가 준비되어 있으며 잔액이 부족한 경우에는 예외처리와 구매한 목록 리스트를 알려준다.

1. 클래스 다이어그램



2. 구현 클래스

◆ Drink 클래스 : 음료수 클래스들의 상위 클래스

- package 명은 com.test06.entity 이다.
- 잔액을 저장하기 위한 price 변수를 선언한다.
- price 변수에 해당하는 getPrice 메서드와 setPrice 메서드를 작성한다.
- price를 인자로 갖는 생성자를 작성한다.

◆ Coffee, Juice, Coke 클래스 : 음료수 클래스

- package 명은 com.test06.entity 이다.
- Drink 클래스를 상속받고 가격을 저장하는 상수값을 설정한다. (실행결과 참조)
- 기본 생성자를 지정하고 super 키워드를 사용하여 상수값을 초기화한다.
- toString() 메서드를 오버라이딩하여 음료수명과 가격을 반환한다. (실행결과 참조)

◆ IVendingMachineBiz 인터페이스: 음료수 자판기 처리 메서드를 선언한 인터페이스

- package 명은 com.test06.biz 이다.
- Class Diagram을 참조하여 추상 메서드를 추가한다.

◆ VendingMachineBiz 클래스 : 음료수 자판기 처리 메서드를 구현한 클래스

- package 명은 com.test06.biz 이다.
- Class Diagram을 참조하여 멤버변수를 정의한다.
(balance: 잔액, 기본값은 1000,
cartList: Drink 타입을 저장할 수 있는 배열 (초기값 3),
count: 현재 배열에 저장된 개수)
- cartDrink()는 음료수 구매 로직을 구현하는 메서드이다
 - 1) 현재 잔액을 출력한다. (Sample Run 참조)
 - 2) 매개변수로 넘겨받은 Drink 타입에 대한 메시지를 출력한다. (실행결과 참조)
메시지 출력시 Drink 타입의 toString() 메소드를 사용한다.
 - 3) 잔액과 구매가격을 비교하여 부족한 경우에 에러 메시지를 출력한다.
 - 4) 현재 갖고 있는 잔액에서 구매가격만큼 뺀다.
 - 5) 배열에 저장된 개수와 배열의 크기가 같으면, 기존의 배열보다 사이즈가 3배인 배열을 생성한 후, 기존의 배열 내용을 새로운 배열에 복사하고 새로운 배열과 기존 배열을 바꾼다.
 - 6) 배열에 구매한 개수만큼 저장하고 count값을 증가시킨다.

-printCart()는 구매 목록 배열에 저장된 정보를 출력하는 메서드이다.

- 1) 배열에 저장된 데이터 타입에 따른 개수를 count한다.
- 2) 배열에 저장된 총 구매금액을 계산한다.
- 3) 구매목록과 남은 잔액을 출력한다.(실행결과 참조)

-printDrinkList()는 음료수 자판기내의 모든 음료수를 출력하는 메서드이다

- 1) drinkList 배열에 저장된 내용을 출력한다. (실행결과 참조)
음료수명은 한글로 출력한다. (예: 커피, 코크, 주스)

2) 마지막에는 잔액을 출력한다. (실행결과 참조)

◆ VendingMachineTest 클래스 : main 클래스

-package명은 com.test06 이다.

-사용자 입력을 받기 위한 메뉴를 화면에 출력한다. (실행결과 참조)

-지정된 메뉴번호 이외의 값dmf 입력하면 예외메시지를 출력한다.(실행결과 참조)

-메뉴번호 9값을 입력하면 종료 메시지를 출력하고 프로그램을 종료한다.

-printMenu() 메서드는 메뉴를 출력하는 메서드이다.

3. 실행 결과

<< 전체 음료수 및 잔액 보기 메뉴 선택한 경우 >>

```
=====
[음료수 자판기 관리 시스템]
=====
1.전체 음료수 및 잔액 보기
2.주스 구입하기 ( 200원 )
3.커피 구입하기 ( 100원 )
4.코크 구입하기 ( 50원 )
5.구입한 음료수 목록 보기
9.종료
```

```
=====
메뉴 입력 => 1
=====
음료수명      가격
=====
커피           100원
코크           50원
주스          200원
-----
현재 잔액: 1000 원
```

<< 주스 구입하기 메뉴 선택한 경우 >>

```
=====
메뉴 입력 => 2
주스를 구입했습니다.현재 잔액: 800 원
```

<< 커피 구입하기 메뉴 선택한 경우 >>

```
=====
메뉴 입력 => 3
커피를 구입했습니다.현재 잔액: 700 원
```

<< 코크 구입하기 메뉴 선택한 경우 >>

```
=====
메뉴 입력 => 4
코크를 구입했습니다.현재 잔액: 650 원
```

<< 구입한 음료수 목록 보기 메뉴 선택한 경우 >>

```
=====
메뉴 입력 => 5
=====음료수 구입 목록=====
주스 : 1 개
```

커피 : 1 개
코크 : 1 개

=====

사용 금액: 350 원
남은 금액: 650 원

<< 잔액이 모자라는 경우 >>

=====

메뉴 입력 => 2
주스를 구입했습니다.현재 잔액: 50 원

=====

[음료수 자판기 관리 시스템]

=====

- 1.전체 음료수 및 잔액 보기
- 2.주스 구입하기 (200원)
- 3.커피 구입하기 (100원)
- 4.코크 구입하기 (50원)
- 5.구입한 음료수 목록 보기
- 9.종료

=====

메뉴 입력 => 2
잔액이 부족하여 주스 구매 불가능합니다.

=====

[음료수 자판기 관리 시스템]

=====

- 1.전체 음료수 및 잔액 보기
- 2.주스 구입하기 (200원)
- 3.커피 구입하기 (100원)
- 4.코크 구입하기 (50원)
- 5.구입한 음료수 목록 보기
- 9.종료

=====

메뉴 입력 => 3
잔액이 부족하여 커피 구매 불가능합니다.

<< 종료 메뉴를 선택한 경우 >>

=====

메뉴 입력 => 9
프로그램 종료

<<메뉴 번호를 잘못 입력한 경우 >>

=====

[음료수 자판기 관리 시스템]

=====

- 1.전체 음료수 및 잔액 보기
- 2.주스 구입하기 (200원)
- 3.커피 구입하기 (100원)
- 4.코크 구입하기 (50원)
- 5.구입한 음료수 목록 보기
- 9.종료

=====

메뉴 입력 => 8
메뉴값을 확인후 다시 입력하세요