

Mini Project Description

Mini project #1

프로젝트 #1



❖ 다음의 정보를 이용하여 Addr 클래스와 AddrMain 클래스를 만들어봅시다.

❖ Addr Class

- 다음의 정보를 저장한다.
 - 이름/전화번호/이메일/주소
 - 그룹(친구 or 가족)
- 다음의 기능을 수행한다.
 - 위 저장 정보를 출력하는 기능 (printInfo())
- 기술 요구사항
 - 변수들은 직접 참조를 막아 캡슐화 처리를 합니다.
 - 변수의 값을 얻을 수 있는 메소드 getter와 변수에 값을 저장할 수 있는 메소드 setter를 정의합니다.
 - 인스턴스 생성과 함께 데이터를 초기화 할 수 있도록 생성자를 정의합니다.

❖ AddrMain Class

- 수행 과정
 - main()메소드를 정의합니다.
 - 연락처 데이터를 저장하는 인스턴스를 생성합니다
 - 생성된 인스턴스의 정보 출력 메소드를 호출합니다.
 - 인스턴스의 각 변수에 값을 바꾸는 메소드를 이용해서 데이터를 수정합니다.
(그룹 정보 변경 : 친구 -> 가족)
 - 인스턴스의 출력 메소드를 다시 실행합니다.

```
이름 : 최은호  
전화번호 : 010-1234-5678  
이메일 : choi@gmail.com  
주소 : 서울  
그룹 : 친구
```

```
-----  
이름 : 최은호  
전화번호 : 010-1234-5678  
이메일 : choi@gmail.com  
주소 : 서울  
그룹 : 가족  
-----
```

Mini project #2

프로젝트 #2

❖ 프로젝트 #1에서 정의한 Addr 클래스를 기반으로 아래의 요구사항을 추가해서 프로그램을 만듭니다.

1. SmartPhone 클래스를 정의합니다(SmartPhone.java). 이 클래스는 연락처 정보를 관리하는 클래스입니다.
 - Addr 클래스의 인스턴스 10개를 저장할 수 있는 배열을 정의합니다.
 - 배열에 인스턴스를 저장하고, 수정하고, 삭제, 저장된 데이터의 리스트를 출력하는 메소드를 정의합니다
2. main()메소드를 아래의 요구조건을 정의해봅시다.(SmartPhoneMain.java)
 - SmartPhone클래스의 인스턴스를 생성합니다.
 - 사용자로부터 입력을 받아 Addr 인스턴스를 생성해서 SmartPhone클래스의 인스턴스가 가지고 있는 배열에 추가합니다. (처음에 연락처 등록 데이터 2개를 입력합니다)
 - 배열의 모든 요소를 출력합니다 (위의 연락처 2개 정보가 출력되는 확인한다.
 - 배열에 연락처 정보를 5개정도 추가해봅니다.
 - 배열의 모든 요소를 출력합니다.
 - 배열의 모든 요소를 검색합니다. (using contentEquals())
 - 첫날은 위의 메소드까지만 개발합니다.
 - 배열의 요소를 삭제해 봅니다.
 - 배열의 요소를 수정해 봅니다.

메소드 와 Console 출력 예시

- ❖ SmartPhone class
 - inputAddrData() //키보드로 부터 입력 받아 객체를 생성함
 - addAddr(Addr Addr) // 배열에 연락처 객체 저장
 - printAddr(Addr Addr) // 객체 정보 출력
 - printAllAddr() //모든 연락처 출력
 - searchAddr(String name) //연락처 검색
 - deleteAddr(String name) // 연락처 삭제
 - editAddr(String name, Addr newAddr) // 연락처 수정
- ❖ SmartPhoneMain class
 - printMenu() //메뉴 정보 출력(1. 연락처 등록 등)

```
Problems Javadoc Declaration Console x
SmartPhoneMain [Java Application] C:\Program Files\J
# 데이터 2개를 입력합니다.
이름 : 김성호
전화번호 : 010-1234-5678
이메일 : shkim@naver.com
주소 : 고양시 삼송
그룹(친구/가족) : 가족
>>> 데이터가 저장되었습니다. (1)
이름 : 홍길동
전화번호 : 010-1111-2222
이메일 : hong@naver.com
주소 : 서울
그룹(친구/가족) : 친구
>>> 데이터가 저장되었습니다. (2)
주소관리 메뉴-----
>> 1. 연락처 등록
>> 2. 모든 연락처 출력
>> 3. 연락처 검색
>> 4. 연락처 삭제
>> 5. 연락처 수정
>> 6. 프로그램 종료
-----
```

Mini project #3

Mini_Project #1-3 Single Linked List

❖ 다음은 연결 자료구조(Linked List)에 대한 내용이다.

❖ 노드

- 연결 자료구조에서 하나의 원소를 표현하기 위한 단위 구조
- <원소, 주소>의 구조



- 데이터 필드(data field)
 - 원소의 값을 저장
 - 저장할 원소의 형태에 따라서 하나 이상의 필드로 구성
- 링크 필드(link field)
 - 다음 노드의 주소를 저장
 - 포인터 변수를 사용하여 주소값을 저장

Console 출력 예시

```
Problems Javadoc Declaration Console x
C:\Program Files\Java\jdk-1.8.0_101\bin> java SingleLinkedList
(1) 공백 리스트에 노드 3개 삽입하기
L = (1, 3, 7)
(2) 3 노드 뒤에 5 노드 삽입하기
L = (1, 3, 5, 7)
(3) 리스트의 노드들 역순으로 바꾸기
L = (7, 5, 3, 1)
(4) 리스트의 마지막 노드 삭제하기
L = (7, 5, 3)
```

Mini project #4

프로젝트 #1-4

❖ 프로젝트 #1-2에서 정의한 Addr 클래스를 상속의 구조로 만들어 봅니다.

1. Addr 클래스는 기본정보를 저장하고, 기본 정보를 출력하는 메소드를 정의합니다.
 - 생성자를 통해 기본 정보들을 초기화합니다.
2. 그룹에 해당하는 정보들을 추가적으로 정의하는 새로운 클래스들을 정의합니다. 회사, 거래처의 정보를 저장하는 하위 클래스를 정의합니다.
 - 회사의 정보를 저장하는 하위클래스를 정의합니다. (CompanyAddr class)
 - 회사이름, 부서이름, 직급 변수 추가
 - 정보를 출력하는 메소드를 오버라이딩해서 추가된 정보를 추가해서 출력
 - 거래처의 정보를 저장하는 하위 클래스를 정의합니다. (CustomerAddr class)
 - 거래처 이름, 거래품목, 직급 변수 추가
 - 정보를 출력하는 메소드를 오버라이딩해서 추가된 정보를 추가해서 출력
3. SmartPhone 클래스의 배열을 다형성의 특징을 이용해서 상위 타입의 배열을 생성해서 하위 클래스의 인스턴스를 저장하는 형태로 프로그램을 작성해 봅니다. (SmartPhone class)
4. SmartPhoneMain class에서는 지난번 코드와 동일하며 PrintMenu는 다음장의 콘솔 예시처럼 작동될 수 있도록 프로그램 해봅시다.

Console 출력 예시

```
Problems Javadoc Declaration Console x
SmartPhoneMain (Java Application) C:\Program Files\Java\
# 데이터 2개를 입력합니다.
이름 : 강성호
전화번호 : 010-5466-5720
이메일 : shkim@naver.com
주소 : 서울
생일 : 0805
그룹 : 회사
회사이름 : 네이버
부서이름 : 개발팀
직급 : 팀장
>>> 데이터가 저장되었습니다. (1)
이름 : 강정호
전화번호 : 010-5466-5720
이메일 : shkimzebra@naver.com
주소 : 서울
생일 : 0806
그룹 : 거래처
거래처이름 : 네이버부동산
품목이름 : 부동산
직급 : 대리
>>> 데이터가 저장되었습니다. (2)
Contact -----
>> 1. 연락처 등록(회사)
>> 2. 연락처 등록(거래처)
>> 3. 모든 연락처 출력
>> 4. 연락처 검색
>> 5. 연락처 삭제
>> 6. 연락처 수정
>> 7. 프로그램 종료
```

Mini project #5

Mini_Project # 1-5

❖ 시나리오

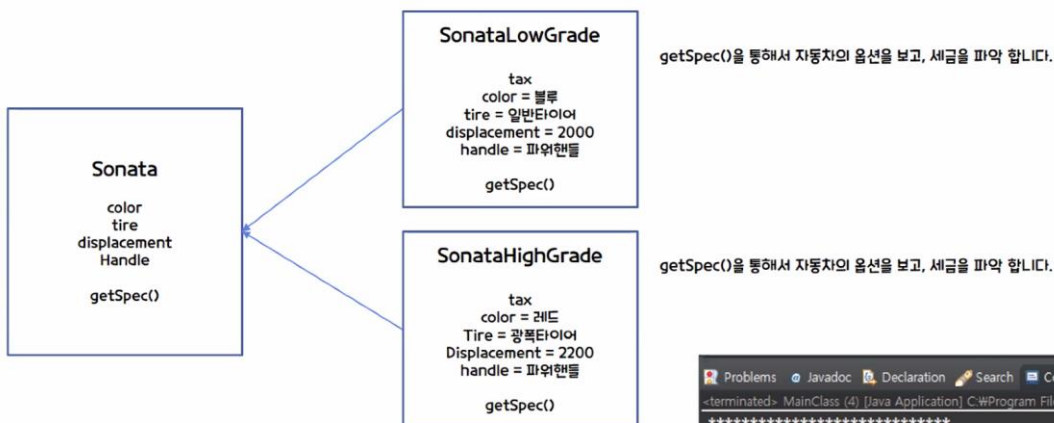
- 자동차의 경우 동일한 자동차 모델에서 옵션 사항에 따라 세금에 차이가 있습니다.
- 배기량이 높은 자동차의 경우에는 세금이 높습니다.
- 클래스 설계를 해 보도록 합니다.

❖ 자동차 스펙

- color (red,blue), tire(normal, wide), displacement(2000,2500), handle(normal, power)

❖ Class 설계

- CarSpecs class : 자동차 스펙에 관한 정보를 갖고있는 클래스
- Sonata class : Sonata 정보에 대한 abstract class
- SonataHighGrade class : high grade에 대한 정보를 갖고 있는 class
- SonataLowGrade class : low grade에 대한 정보를 갖고 있는 class
- MainClass : highGrade와 lowGrade에 대한 객체 생성 및 스펙정보를 호출하는 class



```
Problems Javadoc Declaration Search Console
<terminated> MainClass (4) [Java Application] C:\Program Files\Java\jre1...
*****
색상:블루
타이어:일반타이어
배기량:2000
핸들:파워핸들
세금 :1000
*****
색상:레드
타이어:광폭타이어
배기량:2500
핸들:일반핸들
세금 :1500
```

Mini project #6

Mini_Project # 1-6

❖ 시나리오

S전자에서 다양한 스마트폰을 만듭니다. 모델별 스펙은 아래와 같습니다.

기능	전화 송/수신	접속속도	TV리모컨 기능
a제품	가능	3G	기본 미탑재
b제품	가능	5G	기본 탑재
c제품	가능	4G	기본 미탑재

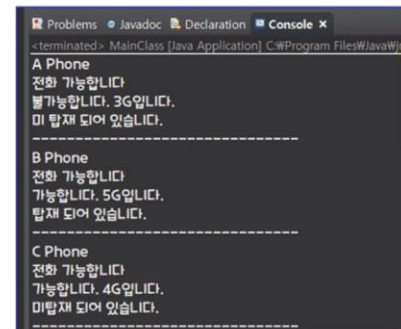
❖ Interface 설계 (IFunction.java)

각 기능으로 인터페이스를 설계한다.

❖ 구현 클래스(APhone/BPhone/CPhone.java)

❖ 결과값(MainClass.java)

각 제품 해당 기능을 콘솔에 Print out 합니다.



```
<terminated> MainClass [Java Application] C:\Program Files\Java\Win
A Phone
전화 가능합니다
불가능합니다. 3G입니다.
미 탑재 되어 있습니다.
-----
B Phone
전화 가능합니다
가능합니다. 5G입니다.
탑재 되어 있습니다.
-----
C Phone
전화 가능합니다
가능합니다. 4G입니다.
미탑재 되어 있습니다.
-----
```

Mini project #7

Ch10-1 Exception handling basic programming

콘솔로부터 태어난 년도를 입력 받을 때 int 타입으로 입력을 받아 정상적으로 숫자를 입력 받았는지 확인하는 프로그램을 작성해봅시다.

사용자 예외 클래스를 정의해서 예외를 발생 시켜 봅시다. (InputException.java)

NumberTest.java에서 main()은 다음의 내용을 실행 합니다.

- 콘솔로 부터 숫자를 입력 받는다.
- 입력 받은 데이터가 숫자가 아닐 경우 "입력하신 데이터는 숫자가 아닙니다"라는 exception 메시지와 원인에 대해 출력이 되도록 코딩해봅시다.

Ch10-2 Exception handling basic programming

콘솔에서 사용자 아이디를 입력 받아 정상적인 영문자의 숫자로만 이루어진 값을 입력 했는지 확인하는 프로그램을 작성해봅시다.

사용자 예외 클래스를 정의해서 예외를 발생 시켜 봅시다. (InputException.java)

InputTest.java에서 main()은 다음의 내용을 실행합니다.

- 콘솔로 부터 아이디를 입력 받는다.
- 입력 받은 아이디가 null, empty인지 확인한다
- 입력 받은 이름이 정상이면 이름을 출력한다.
- 입력 받은 이름이 비정상이면 예외 클래스를 호출하여
"잘못된 이름을 입력했습니다" 라는 메시지가 출력되도록 합니다.

Mini project #8

프로젝트 #1-8

❖ 지금까지 진행한 프로젝트는 배열을 이용해서 연락처를 저장했습니다.

이번 프로젝트에서는 컬렉션 프레임워크를 이용해서 연락처 인스턴스를 저장하도록 프로그램을 만듭시다.

1. List<E>를 이용해서 저장 및 관리하는 프로그램을 만들어 봅시다. (using ArrayList)
2. HashSet<E>을 이용해서 저장 및 관리하는 프로그램을 만들어 봅시다.(using HashSet)
3. HashMap<K,V>를 이용해서 저장 및 관리하는 프로그램을 만들어 봅시다. (using HashMap)
(Key: phoneNumber, Value: Addr)

Mini project #9

프로젝트 #1-9

❖ 연락처 저장 및 불러오기 기능을 만들어 봅시다.

1. File 클래스를 이용해서 저장 폴더 생성
2. 메모리에 저장된 연락처 정보를 파일로 저장하는 기능 (6. 연락처 파일 저장)
3. 파일로 저장된 기능을 불러오는 기능 (7. 연락처 파일로드)
4. 전체 연락처 리스트를 출력하는 기능 실행 (5. 연락처 전체 리스트 보기)