

# 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 (printlnf())
  - 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 : 홍길동
전화번호 : 010-1234-5678
이메일 : abc@gmail.com
주소 : 서울
그룹 : 친구
-----
그룹 정보 변경
-----
이름 : 홍길동
전화번호 : 010-1234-5678
이메일 : abc@gmail.com
주소 : 서울
그룹 : 가족
```



## 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.
    - 배열의 요소를 삭제해 봅니다.
    - 배열의 요소를 수정해 봅니다.



## 메소드 와 Console 출력 예시

- SmartPhone class의 필요한 메소드
  - 키보드로 부터 입력 받아 객체를 생성하는 메소드
  - 배열에 연락처 객체 저장하는 메소드
  - 객체 정보를 출력하는 메소드
  - 모든 연락처 출력하는 메소드
  - 연락처 검색하는 메소드
  - 연락처 삭제하는 메소드
  - 연락처 수정하는 메소드
- SmartPhoneMain class
  - 메뉴 정보 출력(1. 연락처 등록 등)하는 메소드

```
Problems Javadoc Declaration Console x
SmartPhoneMain [Java Application] C:\Program Files\WJ
# 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
-----
```

## 프로젝트 #3

---

- 프로젝트 #2에서 정의한 Addr 클래스를 상속의 구조로 만들어 봅시다.
  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 봅시다. (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\j
# 데이터 2개를 입력합니다.
이름 : 김성호
전화번호 : 010-5466-5720
이메일 : shkim@naver.com
주소 : 삼성
생일 : 0805
그룹 : 회사
회사이름 : 네이버
부서이름 : 개발팀
직급 : 팀장
>>> 데이터가 저장되었습니다. (1)
이름 : 김정호
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
-----
```





## 프로젝트 #6

---

- 프로젝트 #5에서 구현한 구조를 기초로 예외처리를 해봅시다..
  1. 메뉴 입력 시 발생할 수 있는 예외에 대하여 예외처리 합니다.
  2. 연락처 이름 입력 시에 공백에 대한 예외처리와 영문자와 한글만 허용하는 예외처리를 해봅시다.
  3. 전화번호 형식에 맞지 않을 때 예외처리를 하고 중복될 때 예외 상황이 발생하도록 하고 예외 처리를 해봅시다.



## Project #5-1. Selection Sort

### ■ 선택 정렬(selection sort)

- 전체 원소들 중에서 기준 위치에 맞는 원소를 선택하여 자리를 교환하는 방식으로 정렬
- 수행 방법
  - ① 전체 원소 중에서 가장 작은 원소를 찾아서 선택하여 첫 번째 원소와 자리를 교환한다.
  - ② 그 다음 두 번째로 작은 원소를 찾아 선택하여 두 번째 원소와 자리를 교환한다.
  - ③ 그 다음에는 세 번째로 작은 원소를 찾아서 세 번째 원소와 자리를 교환한다.
  - ④ 이 과정을 반복하면서 정렬을 완성한다.

### Project #5-1. Selection Sort

- 앞장의 결과가 아래와 같이 출력되도록 하세요

```
Problems Javadoc Declaration Console x
<terminated> Ex7_1 [Java Application] C:\Program Files\Java\jdk-1
Input Data : 69 10 30 2 16 8 31 22
Sorted Data 1 단계: 2 10 30 69 16 8 31
Sorted Data 2 단계: 2 8 30 69 16 10 31
Sorted Data 3 단계: 2 8 10 69 16 30 31
Sorted Data 4 단계: 2 8 10 16 69 30 31
Sorted Data 5 단계: 2 8 10 16 22 30 31
Sorted Data 6 단계: 2 8 10 16 22 30 31
Sorted Data 7 단계: 2 8 10 16 22 30 31
```

- class Sort와 class Main를 만들어서  
옆의 결과가 나오도록 하세요

## 프로젝트 # 8

---

- 연락처 저장 및 불러오기 기능을 만들어 봅시다.

E c l i p s  
e 파일 없음

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)





## Project #5-2. Bubble Sort

---

- 버블 정렬(bubble sort)

- 인접한 두 개의 원소를 비교하여 자리를 교환하는 방식
  - 첫 번째 원소부터 마지막 원소까지 반복하여 한 단계가 끝나면 가장 큰 원소가 마지막 자리로 정렬
  - 첫 번째 원소부터 인접한 원소끼리 계속 자리를 교환하면서 맨 마지막 자리로 이동하는 모습이 물 속에서 물 위로 올라오는 물방울 모양과 같다고 하여 버블(bubble) 정렬이라 함.

- 수행 과정

- 정렬되지 않은 {69, 10, 30, 2, 16, 8, 31, 22}의 자료들을 버블 정렬 방법으로 정렬하는 과정을 살펴보자.
  - ① 인접한 두 원소를 비교하여 자리를 교환하는 작업을 첫 번째 원소부터 마지막 원소까지 차례로 반복하여 가장 큰 원소 69를 마지막 자리로 정렬

## Project #5-2. Bubble Sort

- 앞장의 결과가 아래와 같이 출력되도록 하세요

```
Problems Javadoc Declaration Console x
<terminated> Ex5_2 [Java Application] C:\Program Files\WJ

Input Data : 69 10 30 2 16 8 31 22

Sorted Data 1 단계:
 10 69 30 2 16 8 31 22
 10 30 69 2 16 8 31 22
 10 30 2 69 16 8 31 22
 10 30 2 16 69 8 31 22
 10 30 2 16 8 69 31 22
 10 30 2 16 8 31 69 22
 10 30 2 16 8 31 22 69

Sorted Data 2 단계:
 10 30 2 16 8 31 22 69
 10 2 30 16 8 31 22 69
 10 2 16 30 8 31 22 69
 10 2 16 8 30 31 22 69
 10 2 16 8 30 31 22 69
 10 2 16 8 30 22 31 69

Sorted Data 3 단계:
 2 10 16 8 30 22 31 69
 2 10 16 8 30 22 31 69
 2 10 8 16 30 22 31 69
 2 10 8 16 30 22 31 69
 2 10 8 16 22 30 31 69
```

```
Sorted Data 4 단계:
 2 10 8 16 22 30 31 69
 2 8 10 16 22 30 31 69
 2 8 10 16 22 30 31 69
 2 8 10 16 22 30 31 69

Sorted Data 5 단계:
 2 8 10 16 22 30 31 69
 2 8 10 16 22 30 31 69
 2 8 10 16 22 30 31 69

Sorted Data 6 단계:
 2 8 10 16 22 30 31 69
 2 8 10 16 22 30 31 69

Sorted Data 7 단계:
 2 8 10 16 22 30 31 69
```

- class Sort와 class Main를 만들어서  
옆의 결과가 나오도록 하세요

## Project #6-1 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

01.03까지

---

- 연습문제 4번

- 주소록 프로그램을 단순 연결 리스트를 이용하여 작성하시오.

작성 조건1)

- 주소록은 이름, 전화번호, 메일 주소를 저장할 수 있도록 노드를 구성하시오.

작성 조건2)

- 새로운 사람에 대한 주소록 삽입 연산을 작성하시오.
- 새로운 노드는 항상 주소록의 마지막에 삽입되도록 하시오.

작성 조건3)

- 지정한 이름에 대한 노드를 주소록에서 검색하여 삭제하는 연산을 작성하시오.