- 인맥관리 프로그램
 - 대학 동창 이름, 전공, 전화번호 정보 저장 및 관리
 - 직장 동료 이름, 부서, 전화번호 정보 저장 및 관리

```
public class UnivFriend {
    private String name;
    private String major;
    private String phone;
    public UnivFriend(String name, String major, String phone) {
        this.name = name;
        this.major = major;
        this.phone = phone;
    }
    public void showInfo() {
        System.out.println("이름:" + this.name);
        System.out.println("전공:" + this.major);
        System.out.println("전광:" + this.phone);
    }
}
```



```
public class CompFriend {
    private String name;
    private String department;
    private String phone;
    public CompFriend(String name, String department, String phone) {
        this.name = name;
        this.department = department;
        this.phone = phone;
    }
    public void showInfo() {
        System.out.println("이름:" + this.name);
        System.out.println("부서:" + this.department);
        System.out.println("전화:" + this.phone);
    }
}
```



```
public class MyFriends {
  public static void main(String[] args) {
      UnivFriend[] ufrns = new UnivFriend[5];
      int ucnt = 0;
      CompFriend[] cfrns = new CompFriend[5];
      int ccnt = 0;
      ufrns[ucnt++] = new UnivFriend("lee", "computer", "01000005555");
      ufrns[ucnt++] = new UnivFriend("seo", "electronics", "01011115555");
      cfrns[ccnt++] = new CompFriend("yoon", "r&d 1", "01022225555");
      cfrns[ccnt++] = new CompFriend("ha", "r&d 2", "01033335555");
      for(int i = 0; i < ucnt; i++) {
         ufrns[i].showInfo();
      for(int i = 0; i < ccnt; i++) {
         cfrns[i].showInfo();
```





- 인스턴스를 저장하는 배열이 두 개
- 대학 동창의 정보를 저장하는 과정과 직장 동료의 정보를 저장하는 과정 분리
- 저장된 정보를 모두 출력할 때 두 개의 반복문 작성
 - -> 특정 이름의 정보를 검색하는 기능을 추가한다면, 두 배열을 모두 검색해야 함
 - -> 만약 저장 대상이 대학 동창과 직장 동료 2개가 아니라 더 세분화되어 많아지면 나뉘는 수 만큼 배열 추가 생성 필요
 - -> 배열의 수가 얼마나 더 늘어날지 모르며, 늘어나는 배열의 수만큼 프로그램 복잡성 증대

● 상속으로 문제 해결

```
public class Friend {
    protected String name;
    protected String phone;
    public Friend(String name, String phone) {
        this.name = name;
        this.phone = phone;
    }
    public void showInfo() {
        System.out.println("이름:" + this.name);
        System.out.println("전화:" + this.phone);
    }
}
```

```
public class UnivFriend extends Friend {
    private String major;
    public UnivFriend(String name, String phone, String major)
{
        super(name, phone);
        this.major = major;
    }
     @Override
    public void showInfo() {
        System.out.println("이름:" + this.name);
        System.out.println("전공:" + this.major);
        System.out.println("전화:" + this.phone);
    }
}
```

```
public class CompFriend extends Friend {
    private String department;
    public CompFriend(String name, String phone, String department) {
        super(name, phone);
        this.department = department;
    }
    @Override
    public void showInfo() {
        System.out.println("이름:" + this.name);
        System.out.println("부서:" + this.department);
        System.out.println("전화:" + this.phone);
    }
}
```

```
public class MyFriends {
    public static void main(String[] args) {
        Friend[] friend = new Friend[10];
        int cnt = 0;
        friend[cnt++] = new UnivFriend("lee", "computer",
"01000005555");
        friend[cnt++] = new UnivFriend("seo", "electronics",
"01011115555");
        friend[cnt++] = new CompFriend("yoon", "r&d 1",
"01022225555");
        friend[cnt++] = new CompFriend("ha", "r&d 2",
"01033335555");
        for(int i = 0; i < cnt; i++) {
            friend[i].showInfo();
        }
    }
}</pre>
```