//消除重复姓名（1 对象）

HashSet<Person> hs = **new** HashSet<Person>();

hs.addAll(listPerson);

listPerson.clear();

listPerson.addAll(hs);

对象：

**public** **class** Person **implements** Parcelable{

**private** String name;

**private** String phone;

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getPhone() {

**return** phone;

}

**public** **void** setPhone(String phone) {

**this**.phone = phone;

}

**public** **int** hashCode() {

**return** name.hashCode();

}

//以名字判断是否为同一对象

**public** **boolean** equals(Object obj) {

**if** (!(obj **instanceof** Person))

**return** **false**;

Person p = (Person) obj;

**return** **this**.name.equals(p.name);

}

/\*\*

\* 序列化实体类

\*/

**public** **static** **final** Parcelable.Creator<Person> *CREATOR* = **new** Creator<Person>() {

@Override

**public** Person[] newArray(**int** arg0) {

**return** **new** Person[arg0];

}

@Override

**public** Person createFromParcel(Parcel arg0) {

Person person = **new** Person();

person.name = arg0.readString();

person.phone = arg0.readString();

**return** person;

}

};

@Override

**public** **int** describeContents() {

**return** 0;

}

/\*\*

\* 将实体类数据写入Parcel

\*/

@Override

**public** **void** writeToParcel(Parcel parcel, **int** flags) {

parcel.writeString(name);

parcel.writeString(phone);

}

}

//消除重复姓名（2 字符串）

String[] n = names.split(";");

String[] p = phones.split(";");

List list = Arrays.asList(n);

List list2 = Arrays.asList(p);

Set set = new HashSet(list);

Set set2 = new HashSet(list2);

String[] newName = (String[]) set.toArray(new String[0]);

String[] newPhone = (String[]) set2.toArray(new String[0]);

names = "";

phones = "";

for(int i=0;i<newName.length;i++){

names = names + newName[i]+";";

phones = phones + newPhone[i]+";";

}