

```
template <class T>
class Set : public List<T> {
public:
    Set(T m):List<T>(m){}
    Set(T m,istream& i):List<T>(m,i){}
    void Union(Set& s1,Set& s2)          // s1 ∪ s2
    {   List<T>::Empty();
        for (s1.First();!s1.IsEol();s1.Next()) Insert(s1.Member());
        for (s2.First();!s2.IsEol();s2.Next()) Insert(s2.Member());
    }
    void Intersection(Set& s1,Set& s2)
    {   List<T>::Empty();
        for (s1.First();!s1.IsEol();s1.Next()) {
            if (s2.IsMember(s1.Member())) Insert(s1.Member());
        }
    }
    void Difference(Set& Minuend,Set& Subtrahend)
    {   List<T>::Empty();
        for (Minuend.First();!Minuend.IsEol();Minuend.Next()) Insert(Minuend.Member());
        Set I(List<T>::MIN); I.Intersection(Minuend,Subtrahend);
        for (I.First();!I.IsEol();I.Next()) Remove(I.Member());
    }
};
```

Figure 1. class Set (array implementation).