

```
#include "List07.h"
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  $\cup$  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)         //s1  $\cap$  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) //Minuend - Subtrahend
    {
        List<T>::Empty();
        Union(Minuend,Subtrahend);
        Set l(List<T>::MIN); l.Intersection(Minuend,Subtrahend);
        for (l.First();!l.IsEol();l.Next()) Remove(l.Member());
    }
};
```

Figure 1. template class *Set* (array implementation).