

```
package weiss.util;
public class HashSet
    extends AbstractCollection implements Set
{
    public HashSet( );
    public HashSet( Collection other );
    public int size( );
    public Object getMatch( Object x );
    public boolean contains( Object x );
    private static boolean isActive
        ( HashEntry [ ] arr, int pos );
    public boolean add( Object x );
    private void rehash( );
    public boolean remove( Object x );
    public void clear( );
    public Iterator iterator( );
    private static class HashEntry
        implements Serializable;
    private int findPos( Object x );
    private void allocateArray( int arraySize );
    private static int nextPrime( int n );
    private static boolean isPrime( int n );
```

```
    private static final int
        DEFAULT_TABLE_SIZE = 101;
    private int currentSize = 0;
    private int occupied = 0;
    private int modCount = 0;
    private HashEntry [ ] array;
}

private class HashSetIterator
    implements Iterator
{
    private int expectedModCount
        = modCount;
    private int currentPos = -1;
    private int visited = 0;
    public boolean hasNext( );
    public Object next( );
    public void remove( );
}
```

```
public class HashMap extends MapImpl
{
    public HashMap( );
    public HashMap( Map other );
    protected Map.Entry makePair( Object key, Object value );
    protected Set makeEmptyKeySet( );
    protected Set clonePairSet( Set pairSet );
    private static final class Pair implements Map.Entry
    {
        public Pair( Object k, Object v )
        public Object getKey( )
        public Object getValue( )
        public int hashCode( )
        public boolean equals( Object other )
        private Object key;
        private Object value;
    }
}
```