**package** generics;

**public** **class** Pair2<T **extends** Comparable>

{

**private** T first;

**private** T second;

**public** T max( )

{

**if** (first.compareTo(second) >= 0)

**return** first;

**else**

**return** second;

}

**public** Pair2()

{

first = **null**;

second = **null**;

}

**public** Pair2(T firstItem, T secondItem)

{

first = firstItem;

second = secondItem;

}

**public** **void** setFirst(T newFirst)

{

first = newFirst;

}

**public** **void** setSecond(T newSecond)

{

second = newSecond;

}

**public** T getFirst()

{

**return** first;

}

**public** T getSecond()

{

**return** second;

}

**public** String toString()

{

**return** ( "[first:" + first.toString() + " second:" + second.toString() +"]");

}

**public** **boolean** equals(Object otherObject)

{

**if** (otherObject == **null**)

**return** **false**;

**else** **if** (getClass( ) != otherObject.getClass( ))

**return** **false**;

**else**

{

Pair2<T> otherPair = (Pair2<T>)otherObject;

**return** (first.equals(otherPair.first)

&& second.equals(otherPair.second));

}

}

**public** **static** **void** main(String[] args)

{

Pair2<String> words = **new** Pair2<String>("chucky", "cheese");

System.***out***.println(words.toString() + " max=" + words.max());

Pair2<Integer> numbers = **new** Pair2<Integer>(50, 21);

System.***out***.println(numbers.toString() + " max=" + numbers.max());

}

}

Output:

[first:chucky second:cheese] max=chucky

[first:50 second:21] max=50