```
    Test.scala ×

1
     package com.crazyjvm.week3
2
3
4
      * Created by chenchao on 14-6-10.
                                                 嵌套函数
5
6
    object LocalFunction extends App{
                                                   Def 里有 def
7
       8
9
10
11
         add2(add2(x,y),z)
12
13
14
15
       println(add3(1,2,3))
16
```

```
ek4 ) (O) Bounds.scala
   1
        package com.crazyjvm.week4
   2
   3
   4
         * Created by chenchao on 14-6-17.
   5
       object Bounds extends App{
   6
   7
       \Theta// val p = new Pair(1,2)
       △// println(p.smaller)
   8
                                                                               P246
   9
          val s1 = new Student("A")
  10
  11
          val s2 = new Student("B")
  12
  13
          val pair1 = new Pair(s1,s2) //Pair[Student,Student]
  14
  15
          val p1 = new Person("AA")
          println(pair1.replaceFirst(p1))
  16
  17
  18
  19
       class Pair[T] (val first : T, val second : T){
          //def smaller = if(first.compareTo(second) < 0) first else second
  20
                                                                               泛型例子2
  21
22
          def replaceFirst[R >: T](newFirst : R) = new Pair(newFirst, second)
  23
       (1)
  24
  25
  26 🔍
        class Person(val name : String)
  27
                                                                               Person 是
  28
         class Student(name : String) extends Person(name)
                                                                               Student 的超类
  29
```

```
6
      object Bounds extends App{
     □// val p = new Pair("A","B")
□// println(p.smaller)
  7
  8
  9
 10
         val s1 = new Student("A")
 11
      12
 13
       // val pairl = new Pair(s1,s2) //Pair[Student,Student]
 14
 15
         val p1 = new Person("AA")
 16
       // println(pair1.replaceFirst(pl))
 17
                                                                           型变
 18
         new X(s1)
 19
         new X(p1)
 20
 21
                                                                            + 协变
      (a)
 22
                                                                            - 逆变
 23
      □class Pair[T < Comparable[T]] (val first : T, val second : T){
 24
         def smaller = if(first.compareTo(second) < 0) first else second</pre>
 25
         //def\ replaceFirst[R >: T] (newFirst : R) = new\ Pair(newFirst, second)
 26
 27
      9}
 28
 29
 30
       class Person(val name : String)
 31
       class Student(name : String) extends Person(name)
 32
 33
 34
      白//T <% V ----> T到V的隐式转换
     ⊖//[T:M] ---> M[T]
 35
M36
 37
     ⊖class X[-T](val x : T){
 38
 39
 40
      (a)
 41
```

隐式转换

```
K4 / U ImplicitDemo.scala
   Function1.scala × 0 Bounds.scala ×
                                     1
        package com.crazyjvm.week4
  2
  3
      import java.io.File
  4
      ⊕import scala.io.Source
  5
  6
  7
        * Created by chenchao on 14-6-17.
  8
  9
      pobject ImplicitDemo extends App{
 10
 11
          import Context.
                                                                        Import ∼
         new File("xxx"). read
 12
 13
      9}
 14
 15
 16
      class RichFile(val file : File){
 17
         def read = Source.fromFile(file.getPath).mkString
 18
 19
 20
      object Context{
 21
         implicit def file2RichFile(f : File) = new RichFile(f)
      台}
 22
```

```
Function1.scala ×
                    O Bounds.scala ×
                                       @ ImplicitDemo.scala ×
                                                              In Compar
12
       // new File("xxx").read
13
         //import Context.
         //AAA.print("Jack")
14
15

△ //println(1.add2)

16
17
         val p1 = new Pair2(1,2)
18
         val p2 = new Pair2("A","B")
19
                                                                        上下文界定1
20
         println(pl.smaller)
         println(p2.smaller)
21
22
                                                                          P330
23
24
     Dclass Pair2[T : Ordering](val first : T, val second : T){
25
         def smaller(implicit ord : Ordering[T]) =
26
           if (ord.compare(first, second) < 0 ) first else second
     0}
27
28
29
     class Line(val len : Double){
30 0
         override def toString() = "Length of lone : " + len
31
32
```

```
ek4 0 ImplicitDemo.scala
   Ta Function1.scala × @ Bounds.scala ×

    ImplicitDemo.scala ×

                                                                 1a Comparable
  11
        ⊕// import Context._
  12
         // new File("xxx").read
  13
           //import Context.
  14
           //AAA.print("Jack")
  15
        16
  17
           implicit object Line extends LineOrdering
                                                                               上下文界定2
  18
           val pl = new Pair2(1,2)
  19
           val p2 = new Pair2("A","B")
  20
  21
           val 11 = new Line(1)
  22
           val 12 = new Line(2)
  23
  24
           val p3 = new Pair2[Line](l1, l2)
  25
  26
           println(pl.smaller)
  27
           println(p2.smaller)
           println(p3.smaller)
  28
  29
  30
  31
        class Pair2[T : Ordering](val first : T, val second : T){
  32
           def smaller(implicit ord : Ordering[T]) =
  33
             if (ord.compare(first, second) < 0 ) first else second
        白}
  34
  36 % ctrait LineOrdering extends Ordering[Line]{
  35
           override def compare(x : Line, y : Line) = {
  if(x.len < y.len) -1</pre>
  38
             else if(x.len = y.len) 0
  39
  40
             else 1
  41
  42
  43
  44
        class Line(val len : Double){
IM 45 of
           override def toString() = "Length of line : " + len
  46
        △}
  47
  48
```