**import** org.apache.hadoop.hbase.HBaseConfiguration  
**import** org.apache.hadoop.hbase.mapreduce.TableInputFormat  
**import** org.apache.hadoop.hbase.util.Bytes  
**import** org.apache.spark.rdd.RDD  
**import** org.apache.spark.{SparkConf, SparkContext}  
  
*/\*\*  
 \* Created by Administrator on 2017/2/9 0009.  
 \*/***object** CreateRdd {  
 **def** main(args: Array[String]): Unit = {  
 **val** conf = **new** SparkConf().setAppName(**"CreateRdd"**).setMaster(**"local"**)  
 **val** sc: SparkContext = **new** SparkContext(conf)  
  
 *//1.并行化集合创建，主要是用来通过少量数据测试编程逻辑* **val** data = *Array*(1,2,3,4,5)  
 *//第二个参数是切片数量  
// val rdd: RDD[Int] = sc.parallelize(data)* **val** rdd: RDD[Int] = sc.parallelize(1 to 5,3)  
  
 *//2.外部数据集  
 //本地文件提交到spark集群，确保每个节点在指定目录都存在该文件  
 //textFile 默认指定的是hdfs路径  
// val file: RDD[String] = sc.textFile("file:\\F:\\vm\_linux\\bigdata\\vmware.log")  
// val file: RDD[String] = sc.textFile("hdfs:///mydata/data.txt")  
// file.foreach(println)  
  
 //hbase数据的RDD创建* **val** config = HBaseConfiguration.*create*()  
 config.set(TableInputFormat.*INPUT\_TABLE*, **"iemployee"**)  
  
 **val** hbaseRdd = sc.newAPIHadoopRDD(config, *classOf*[TableInputFormat],  
 *classOf*[org.apache.hadoop.hbase.io.ImmutableBytesWritable],  
 *classOf*[org.apache.hadoop.hbase.client.Result])  
  
*// println(hbaseRdd.count())* hbaseRdd.foreach{**case** (\_,result) =>{  
 *//获取行键* **val** key = Bytes.*toString*(result.getRow)  
 *//通过列族和列名获取列* **val** insuranceDental = Bytes.*toString*(result.getValue(Bytes.*toBytes*(**"insurance"**),Bytes.*toBytes*(**"dental"**)))  
 **val** skillsManagement = Bytes.*toString*(result.getValue(Bytes.*toBytes*(**"skills"**),Bytes.*toBytes*(**"management"**)))  
 *println*(**"Row key:"**+key+**" insuranceDental: "**+insuranceDental+**" skillsManagement: "**+skillsManagement)  
 }}  
 }  
}