**import** java.util.Properties  
  
**import** org.apache.log4j.{Level, Logger}  
**import** org.apache.spark.rdd.RDD  
**import** org.apache.spark.sql.{DataFrame, Row, SQLContext}  
**import** org.apache.spark.sql.types.{DataType, DataTypes, StructField, StructType}  
**import** org.apache.spark.{SparkConf, SparkContext}  
**import** scala.collection.JavaConverters.\_  
  
*/\*\*  
 \* Created by Administrator on 2017/4/25 0025.  
 \*/***case class** Person(name:String, age:java.lang.Long)  
  
**object** CreateDF {  
 **def** main(args: Array[String]): Unit = {  
 Logger.*getLogger*(**"org.apache.spark"**).setLevel(Level.*WARN*)  
 Logger.*getLogger*(**"org.eclipse.jetty.server"**).setLevel(Level.*OFF*)  
 *//spark入口  
 //spark conf配置对象* **val** conf = **new** SparkConf().setAppName(**"CreateDF"**).setMaster(**"local[2]"**)  
 **val** sc = **new** SparkContext(conf)  
 *//Spark SQL的入口* **val** sqlContext = **new** SQLContext(sc)  
 *//一. DataFrame创建  
// 1.json文件  
// val df = sqlContext.read.json("file:\\G:\\code\\source\_code\\spark\\examples\\src\\main\\resources\\people.json")  
// val df = sqlContext.read.format("json").load("G:\\code\\source\_code\\spark\\examples\\src\\main\\resources\\people.json")  
 // sqlContext.read.format("json").load("file:\\G:\\code\\source\_code\\spark\\examples\\src\\main\\resources\\people.json")  
 // 2.parquet文件  
// val df = sqlContext.read.parquet("file:\\G:\\code\\source\_code\\spark\\examples\\src\\main\\resources\\users.parquet")  
// 3.jdbc方式创建  
// val props = new Properties()  
// props.put("user","root")  
// props.put("password","123456")  
// val df = sqlContext.read.jdbc("jdbc:mysql://hdp1:3306/spark","student",props)  
// 4.通过表创建  
// df.registerTempTable("student")  
// var sql = sqlContext.sql("select \* from student")  
// sql.printSchema()  
// sql.show()  
// 5.avro文件创建  
// import com.databricks.spark.avro.\_  
// val df = sqlContext.read.avro("D:/code/spark\_code/course/data/users.avro")  
// 6.通过RDD的方式  
// 6.1反射方式创建DataFrame  
// import sqlContext.implicits.\_  
// var rdd = sc.textFile("file:\\G:\\code\\source\_code\\spark\\examples\\src\\main\\resources\\people.txt")  
// .map { line =>  
// val strs = line.split(",")  
// Person(strs(0), strs(1).trim.toInt)  
// }  
// val df = rdd.toDF()  
// 6.2注册元数据方法  
// var rdd = sc.textFile("file:\\G:\\code\\source\_code\\spark\\examples\\src\\main\\resources\\people.txt")  
// .map { line =>  
// val strs = line.split(",")  
// Row(strs(0), strs(1).trim.toInt)  
// }  
// var structType = StructType(Array(  
// StructField("name", DataTypes.StringType),  
// StructField("age", DataTypes.IntegerType)  
// ))  
// val df = sqlContext.createDataFrame(rdd,structType)  
  
 //df操作  
// df.printSchema() //打印对应的约束信息  
// df.show() //小数据量时候，客户端显示数据  
// val arrs = df.collect()  
// val list = df.collectAsList()  
// for(i <- 0 until list.size()){  
// println(list.get(i))  
// }  
// for(ele <- list){  
// println(ele)  
// }  
// println(df.count())  
// println(df.describe("name","age"))  
// println(df.first())  
// for(ele <- df.head(2) ){  
// println(ele)  
// }  
// for(ele <- df.take(1)){  
// println(ele)  
// }  
// for(ele <- df.columns){  
 // println(ele)  
 // }  
// println(df.schema)  
// println(df.select("age").explain())  
// 条件过滤  
// println(df.filter(df.col("age").gt(20)).first())  
// println(df.filter(df.col("age") > 20).first())  
  
// println(df.agg(("name" -> "count")).first())  
// println(df.groupBy("name").count())  
  
// df.registerTempTable("people")  
// println(sqlContext.sql("select \* from people where age > 20").first())  
  
// 第二部分：Dataset  
// 1.Dataset的创建  
// import sqlContext.implicits.\_  
// var dS = List(Person("Kevin",24),Person("Jhon",20)).toDS()  
// val list = List(Person("Kevin",24),Person("Jhon",20))  
// val frame = sqlContext.createDataFrame(list)  
// frame.printSchema()  
  
// var rdd = sc.textFile("file:\\G:\\code\\source\_code\\spark\\examples\\src\\main\\resources\\people.txt")  
// .map { line =>  
// val strs = line.split(",")  
// Person(strs(0), strs(1).trim.toInt)  
// }  
// var dS = rdd.toDS()  
// dS.printSchema()* **import** sqlContext.implicits.\_  
 **var** frame = sqlContext.read.json(**"file:\\G:\\code\\source\_code\\spark\\examples\\src\\main\\resources\\people.json"**)  
*// frame.printSchema()* **var** dataset = frame.as[Person]  
*// dataset.printSchema()  
// dataset.show()  
// dataset.filter(person => person.age > 21).show()  
 println*(dataset.groupBy(person => person.name).count().show())  
  
 }  
}