Java 核心技术(进阶)

第四章 高级文件处理

第三节 JSON简介及解析

华东师范大学 陈良育

JSON概念

- JSON
  - JavaScript Object Notation, JS 对象表示法
  - 是一种轻量级的数据交换格式
  - 类似XML，更小、更快、更易解析
  - 最早用于Javascript中，容易解析，最后推广到全语言
  - 尽管使用Javascript语法，但是独立于编程语言

JSONObject和JSONArray

- 名称/值对。如"firstName":"John"
  - JSON对象：{“name":"Jo","email":"a@b.com"}
  - 数据在键值对中
  - 数据由逗号分隔
  - 花括号保存对象
- JSON数组
  - 方括号保存数组

  [{“name":"Jo","email":"a@b.com"}, {“name":"Jo","email":"a@b.com"}]

Java的JSON处理

- org.json：JSON官方推荐的解析类
  - 简单易用，通用性强
  - 复杂功能欠缺
- GSON：Google出品
  - 基于反射，可以实现JSON对象、JSON字符串和Java对象互转
- Jackson：号称最快的JSON处理器
  - 简单易用，社区更新和发布速度比较快

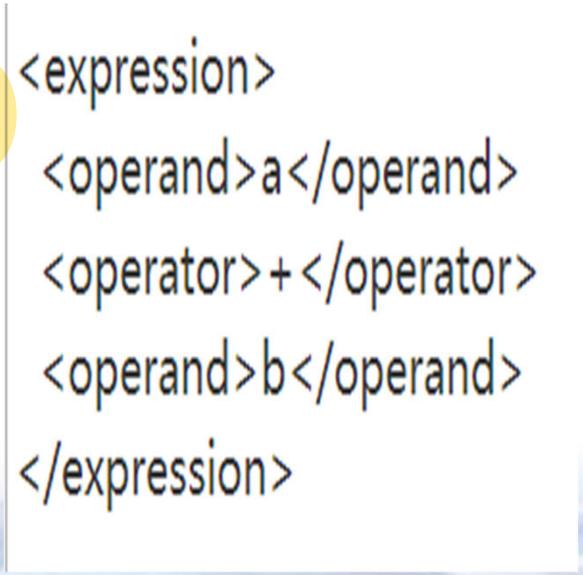
JSON 主要用途

- JSON生成
- JSON解析
- JSON校验
- 和Java Bean对象进行互解析
  - 具有一个无参的构造函数
  - 可以包括多个属性，所有属性都是private
  - 每个属性都有相应的Getter/Setter方法
  - Java Bean用于封装数据，又可称为POJO(Plain Old Java Object)

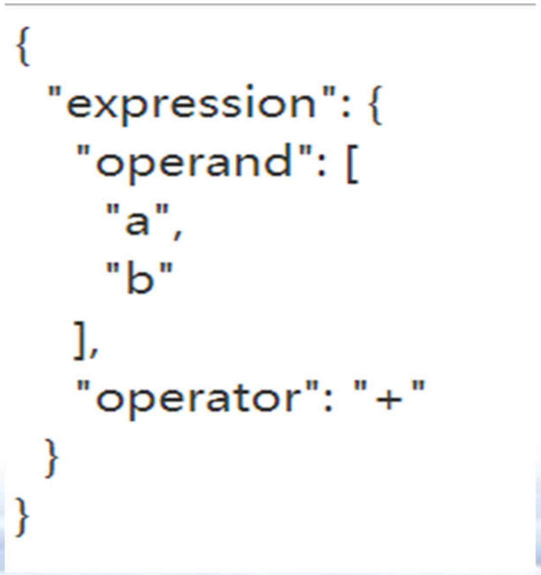
JSON和XML比较

- 都是数据交换格式，可读性强，可扩展性高
- 大部分的情况下，JSON更具优势（编码简单，转换方便），而且JSON字符长度一般小于XML，传输效率更高

- XML更加注重标签和顺序
- JSON会丢失信息



```xml
<expression>
 <operand>a</operand>
 <operator>+</operator>
 <operand>b</operand>
</expression>
```

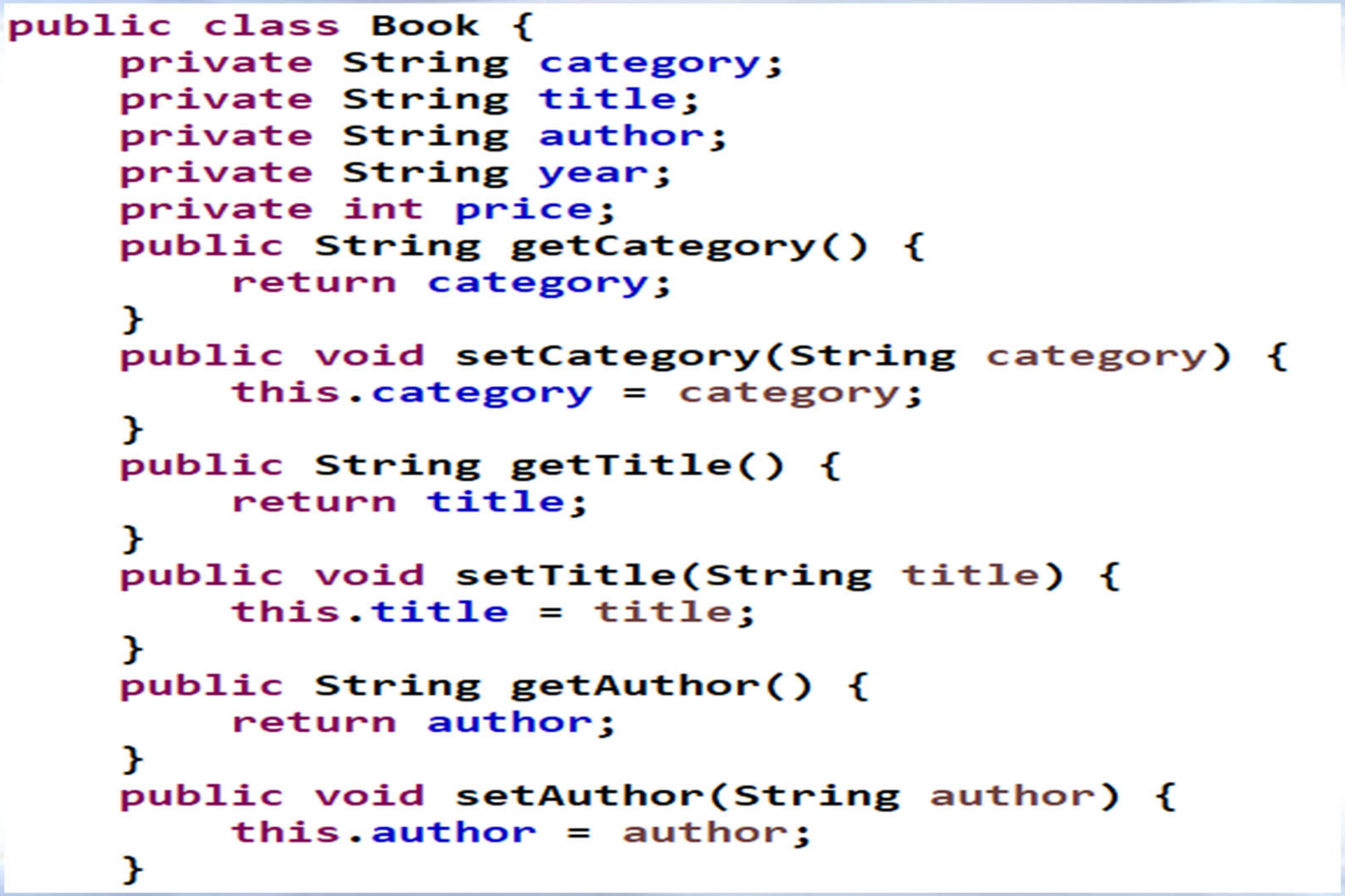


```json
{
  "expression": {
   "operand": [
    "a",
    "b"
   ],
   "operator": "+"
  }
}
```

总结

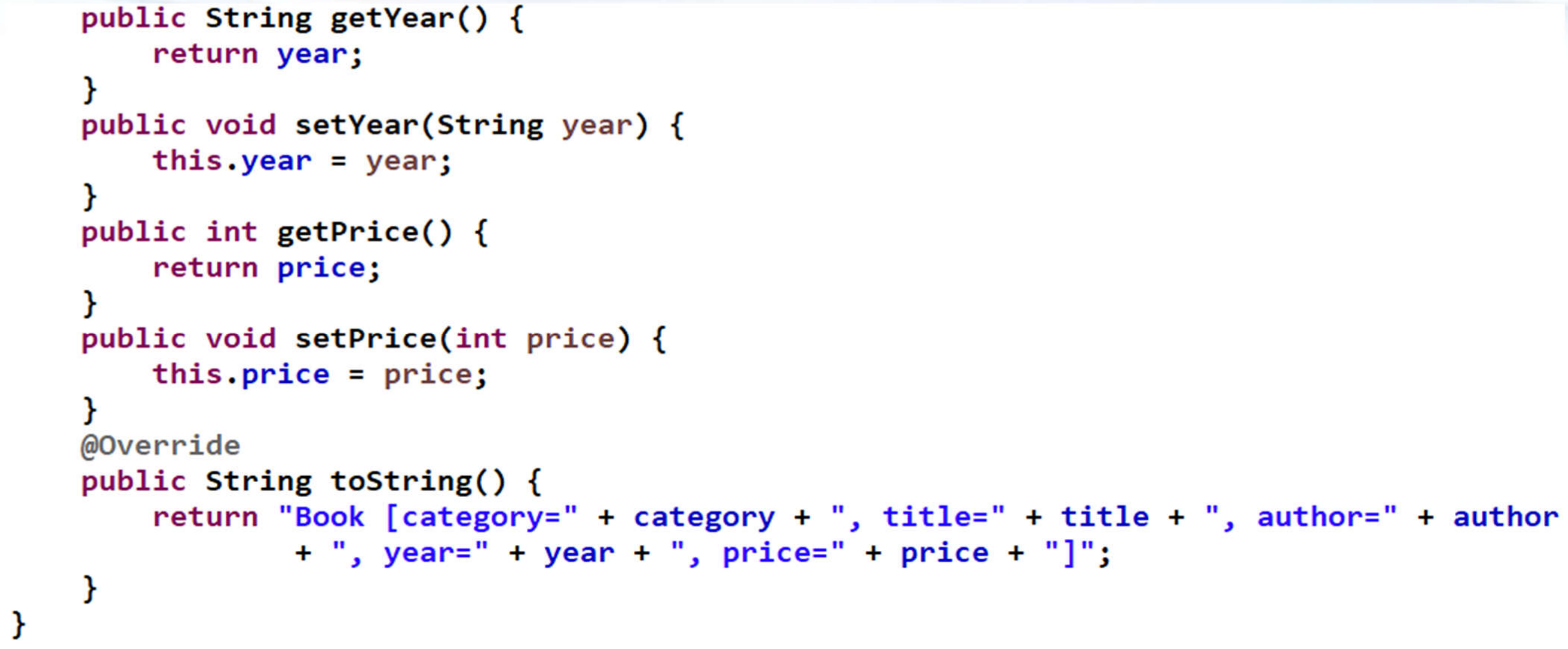
- JSON是一种独立于编程语言的、轻量的、数据交换格式
- 有多种第三方库辅助我们进行JSON生成和解析
- 注意：JSON会丢失顺序性。

代码(1) Book.java



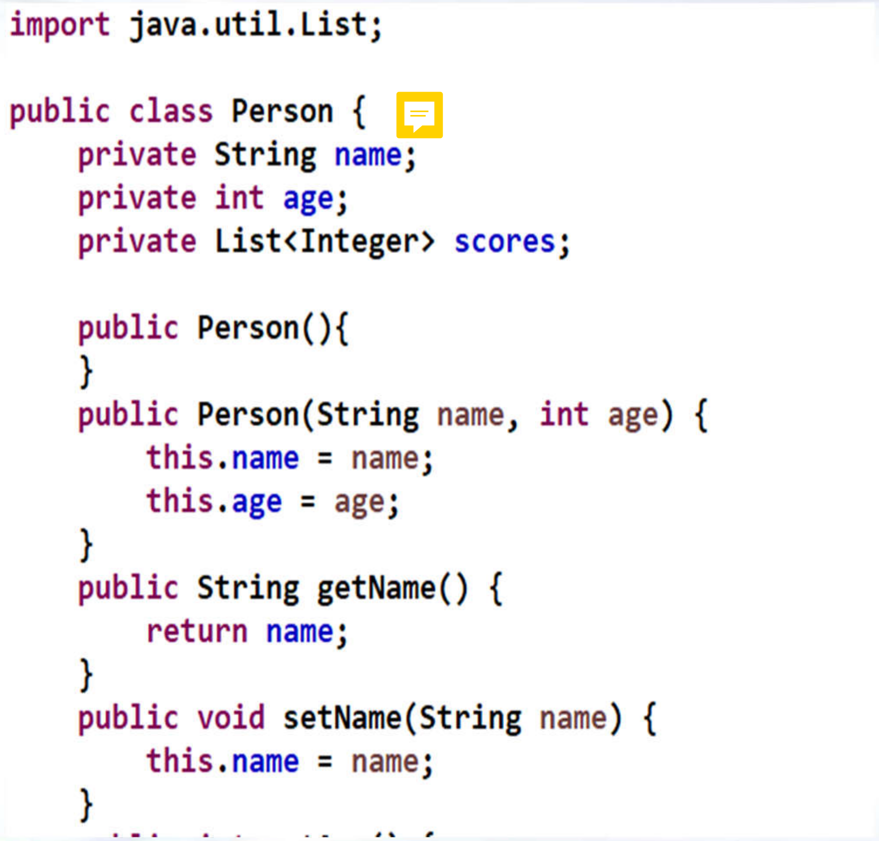
```java
public class Book {
    private String category;
    private String title;
    private String author;
    private String year;
    private int price;
    public String getCategory() {
        return category;
    }
    public void setCategory(String category) {
        this.category = category;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getAuthor() {
        return author;
    }
    public void setAuthor(String author) {
        this.author = author;
    }
```

代码(2) Book.java

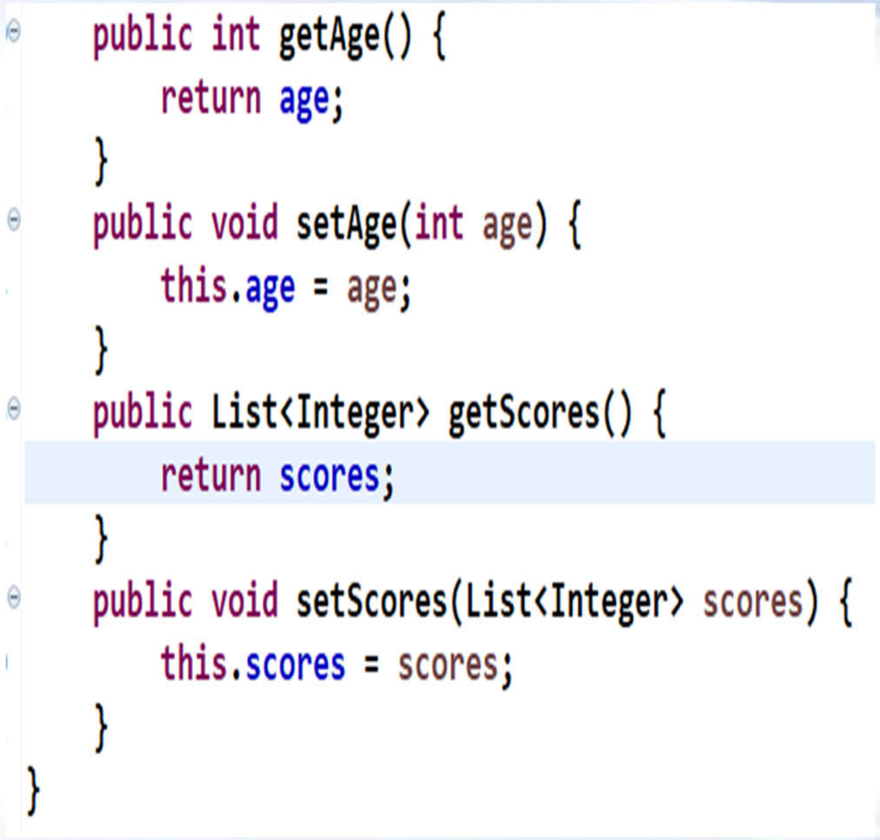


```java
    public String getYear() {
        return year;
    }
    public void setYear(String year) {
        this.year = year;
    }
    public int getPrice() {
        return price;
    }
    public void setPrice(int price) {
        this.price = price;
    }
    @Override
    public String toString() {
        return "Book [category=" + category + ", title=" + title + ", author=" + author
                + ", year=" + year + ", price=" + price + "]";
    }
}
```

代码(3) Person.java

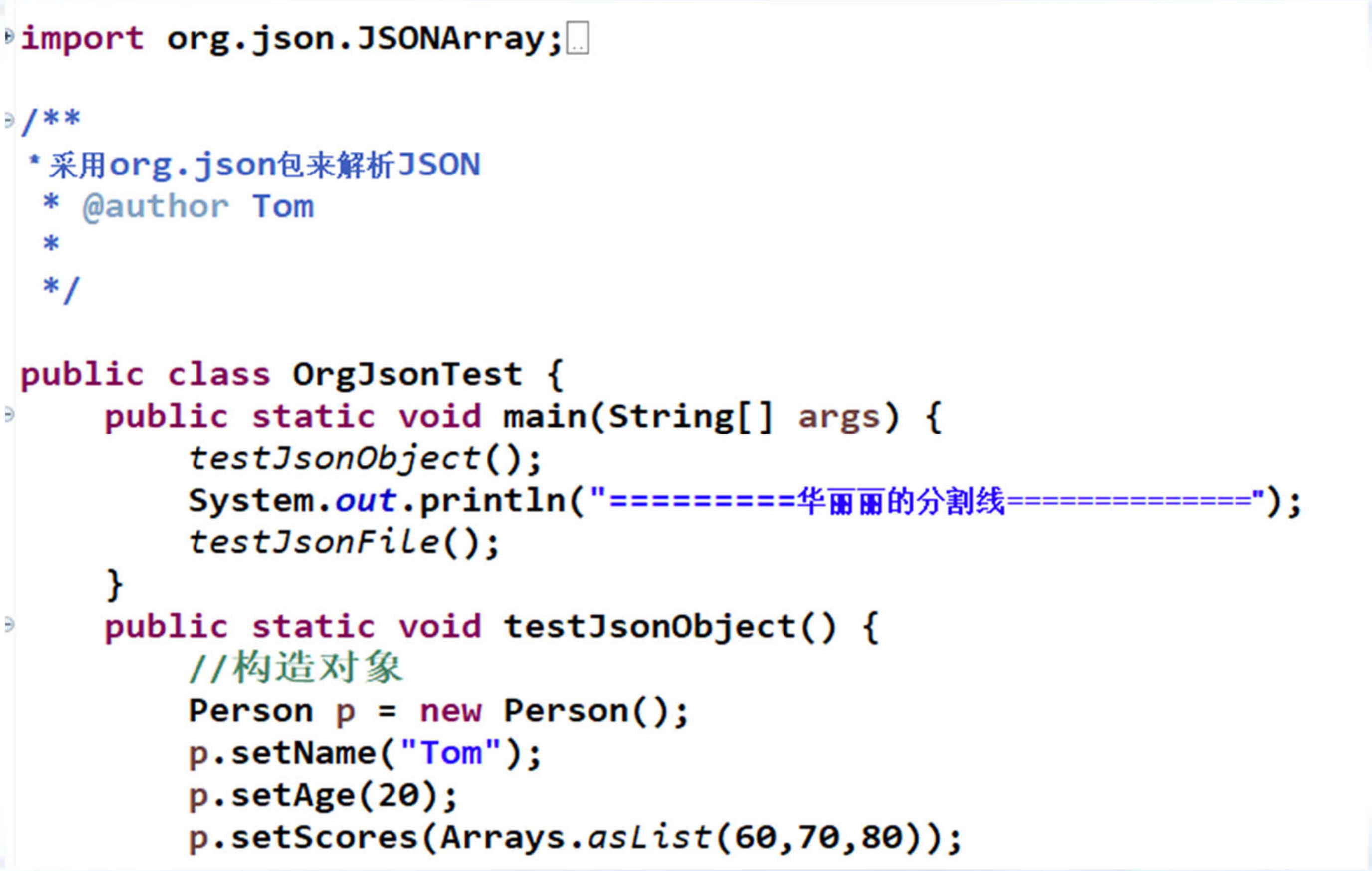


```java
import java.util.List;

public class Person {
    private String name;
    private int age;
    private List<Integer> scores;

    public Person(){
    }
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
```



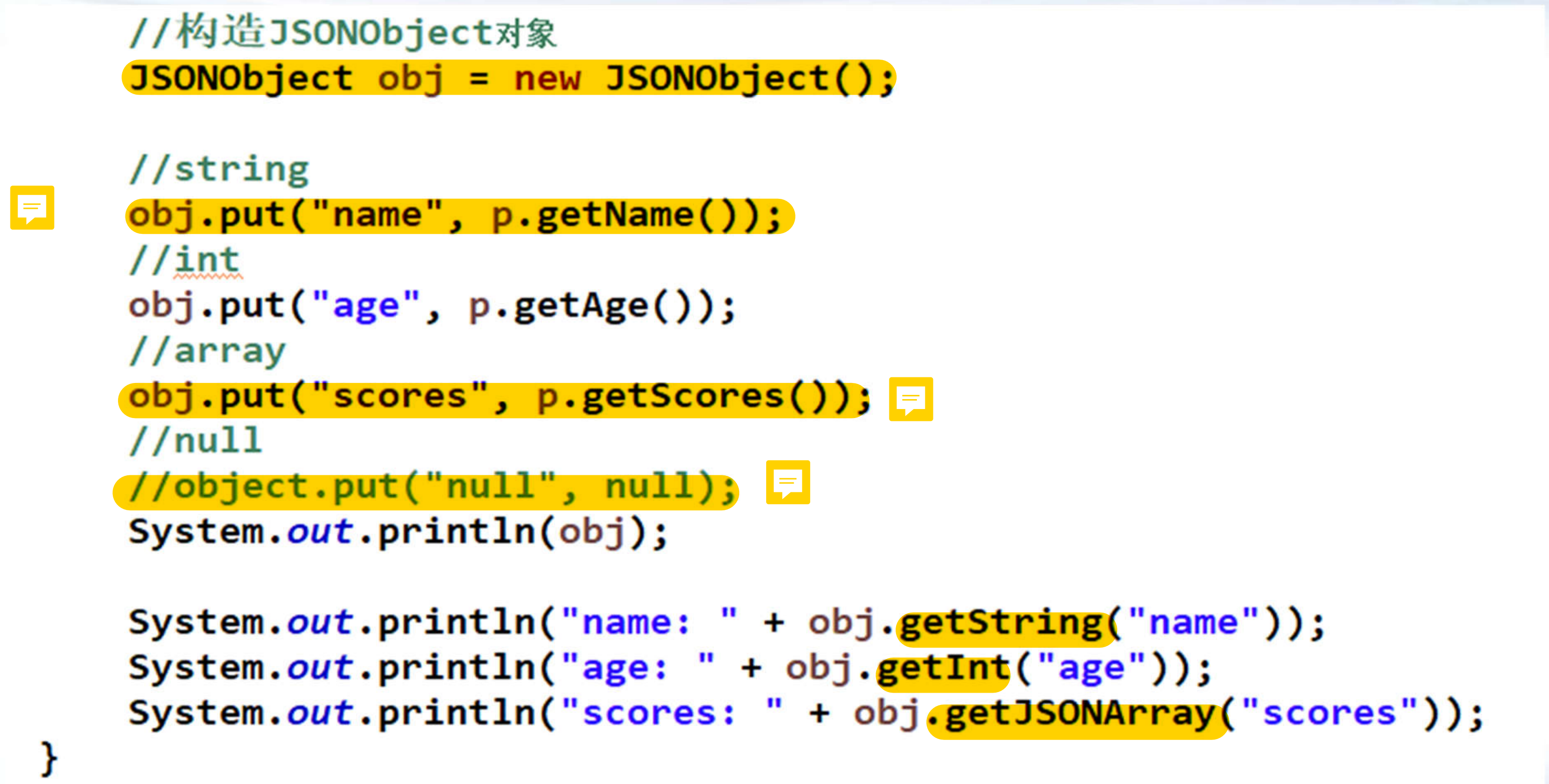
```java
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public List<Integer> getScores() {
        return scores;
    }
    public void setScores(List<Integer> scores) {
        this.scores = scores;
    }
}
```

代码(4) OrgJsonTest.java



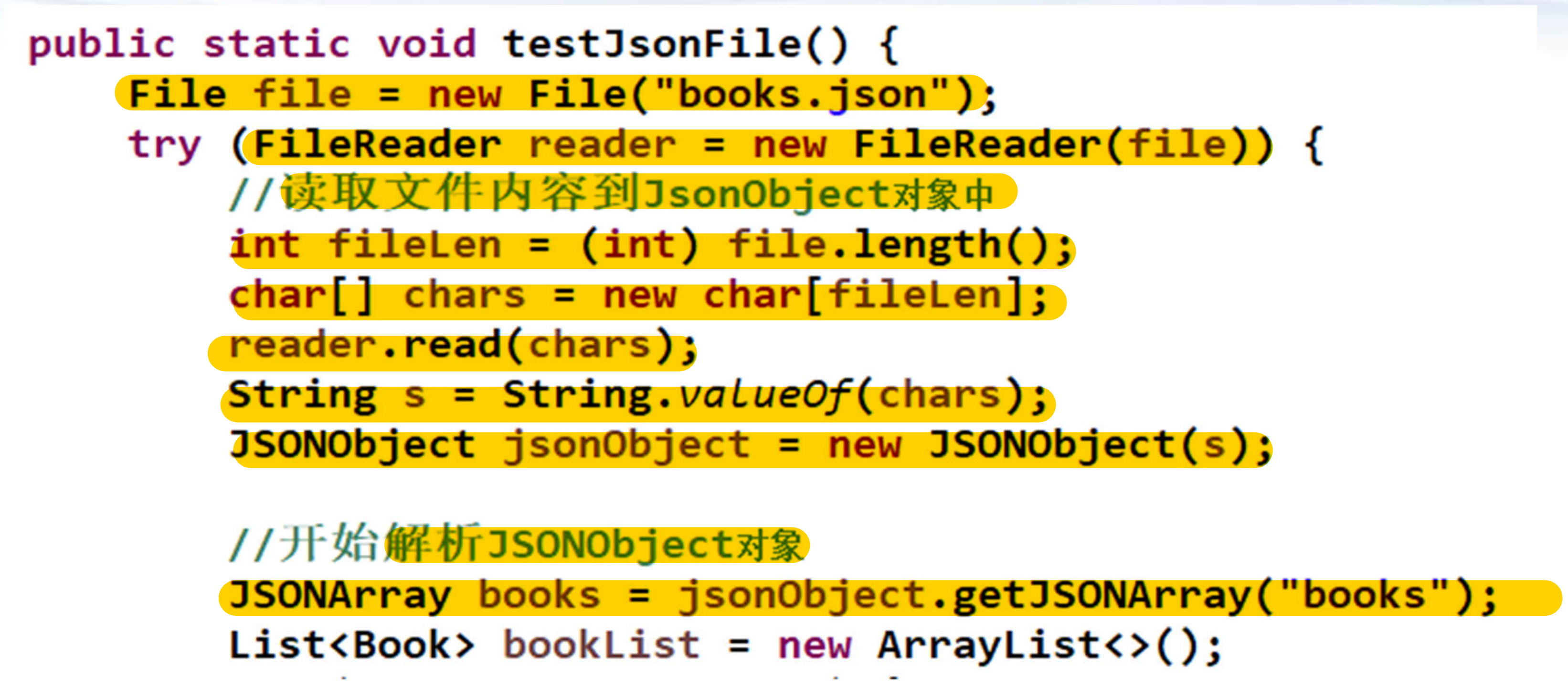
```
import org.json.JSONArray;

/**
 *采用org.json包来解析JSON
 * @author Tom
 *
 */

public class OrgJsonTest {
    public static void main(String[] args) {
        testJsonObject();
        System.out.println("==========华丽丽的分割线==============");
        testJsonFile();
    }
    public static void testJsonObject() {
        //构造对象
        Person p = new Person();
        p.setName("Tom");
        p.setAge(20);
        p.setScores(Arrays.asList(60,70,80));
```

代码(5) OrgJsonTest.java



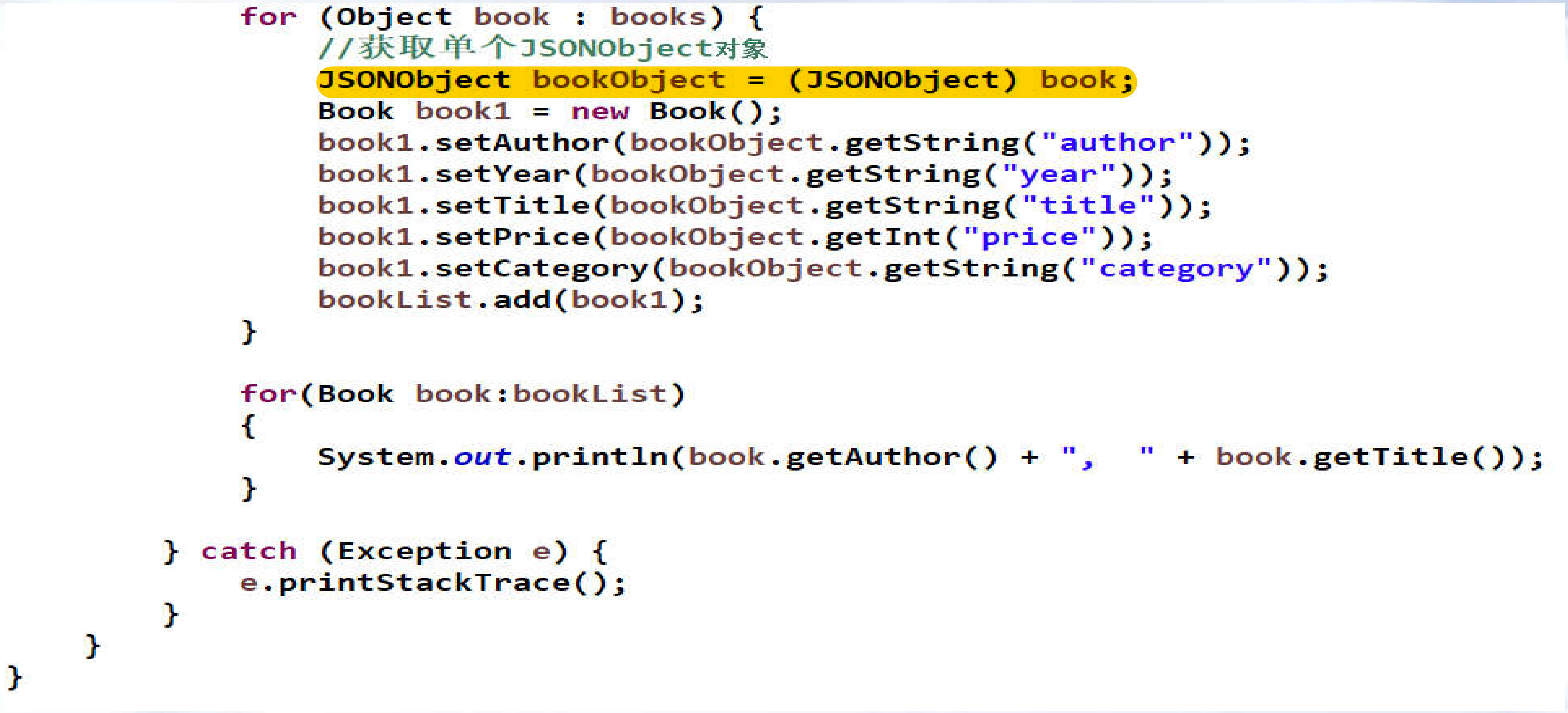
```
        //构造JSONObject对象
        JSONObject obj = new JSONObject();

        //string
        obj.put("name", p.getName());
        //int
        obj.put("age", p.getAge());
        //array
        obj.put("scores", p.getScores());
        //null
        //object.put("null", null);
        System.out.println(obj);

        System.out.println("name: " + obj.getString("name"));
        System.out.println("age: " + obj.getInt("age"));
        System.out.println("scores: " + obj.getJSONArray("scores"));
    }
```

代码(6) OrgJsonTest.java



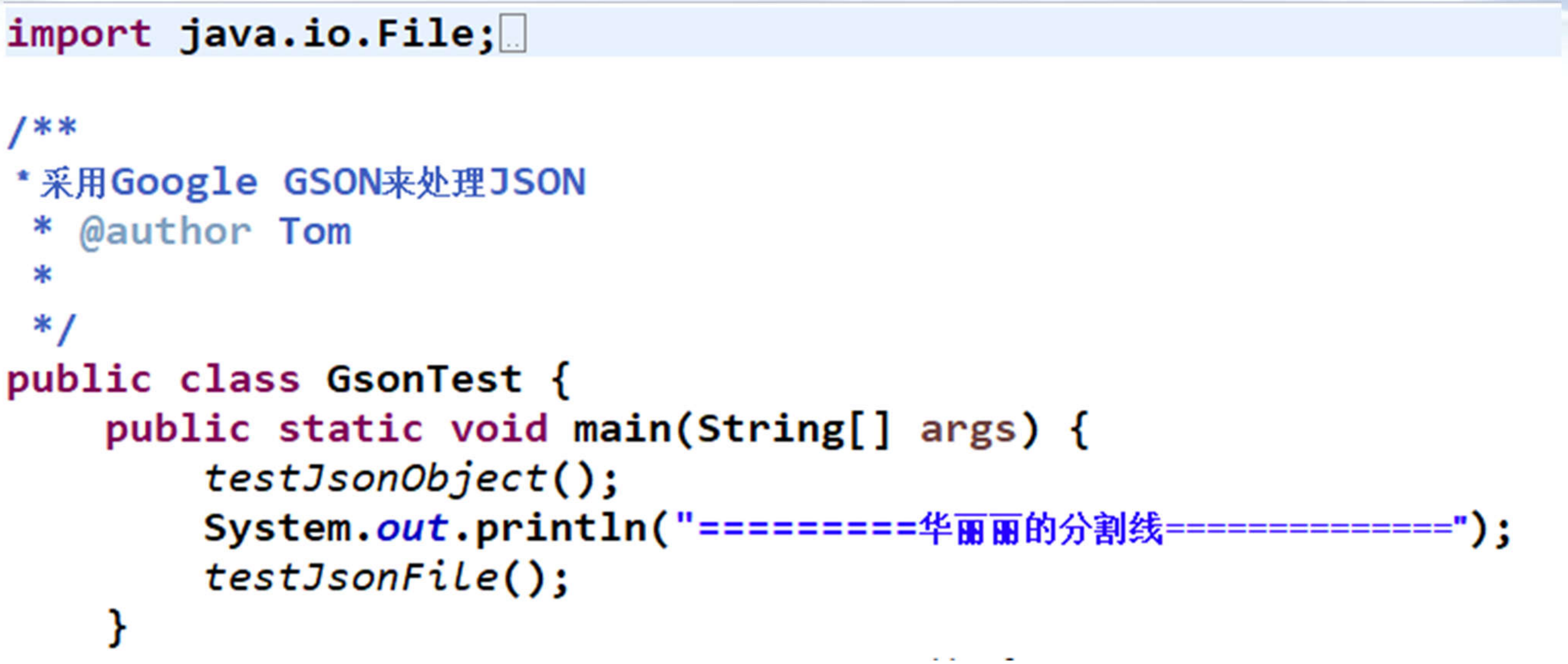
```java
public static void testJsonFile() {
    File file = new File("books.json");
    try (FileReader reader = new FileReader(file)) {
        //读取文件内容到JsonObject对象中
        int fileLen = (int) file.length();
        char[] chars = new char[fileLen];
        reader.read(chars);
        String s = String.valueOf(chars);
        JSONObject jsonObject = new JSONObject(s);

        //开始解析JSONObject对象
        JSONArray books = jsonObject.getJSONArray("books");
        List<Book> bookList = new ArrayList<>();
```

代码(7) OrgJsonTest.java



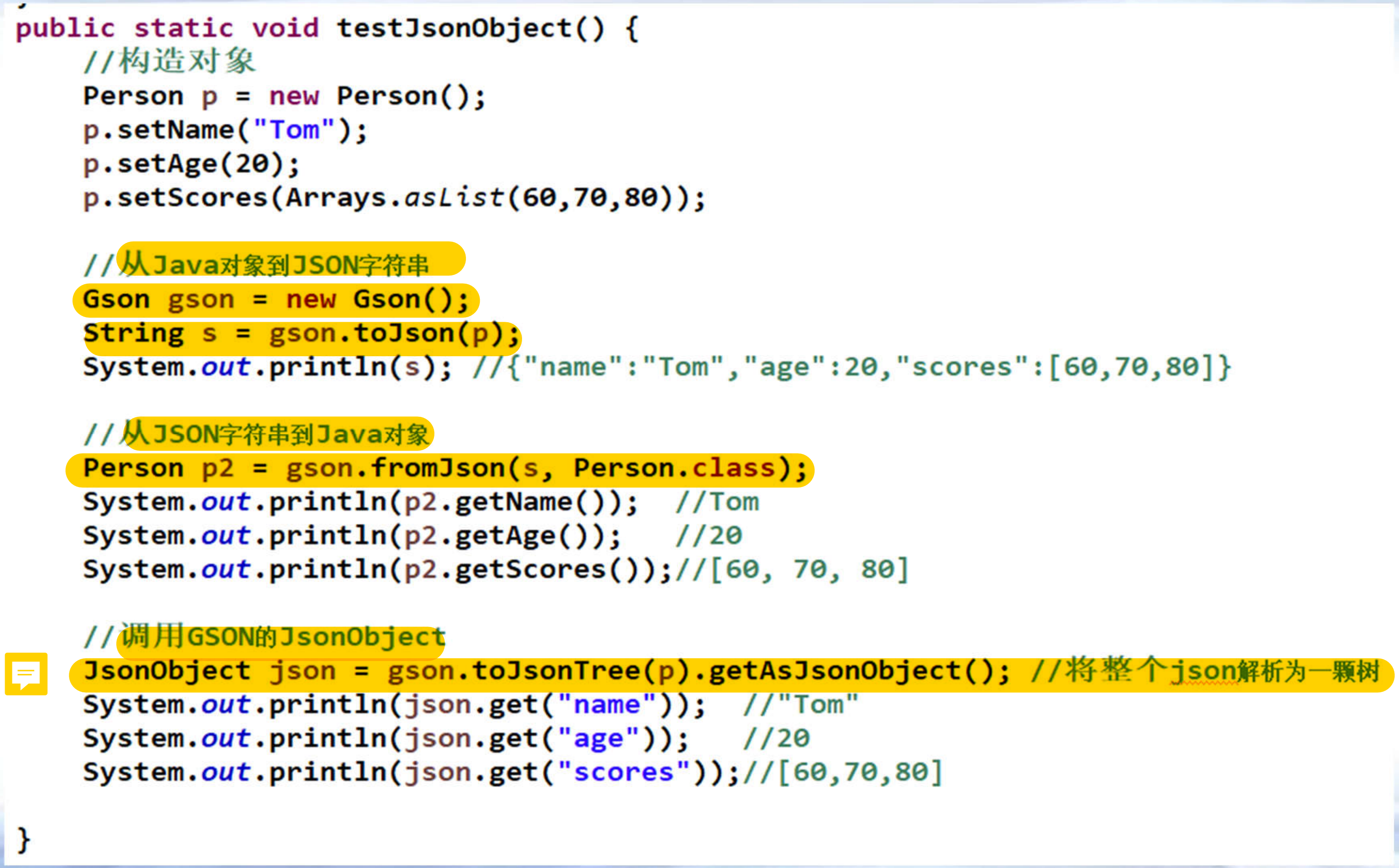
```java
            for (Object book : books) {
                //获取单个JSONObject对象
                JSONObject bookObject = (JSONObject) book;
                Book book1 = new Book();
                book1.setAuthor(bookObject.getString("author"));
                book1.setYear(bookObject.getString("year"));
                book1.setTitle(bookObject.getString("title"));
                book1.setPrice(bookObject.getInt("price"));
                book1.setCategory(bookObject.getString("category"));
                bookList.add(book1);
            }

            for(Book book:bookList)
            {
                System.out.println(book.getAuthor() + ",  " + book.getTitle());
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

代码(8) GsonTest.java



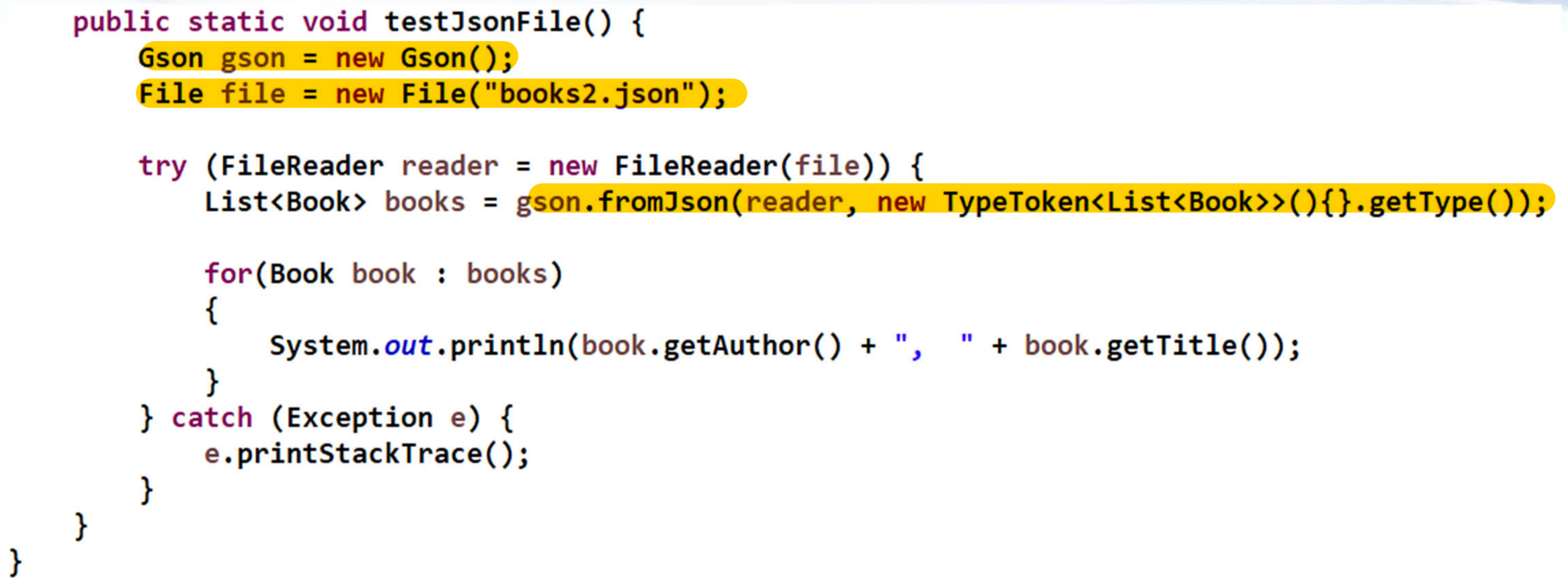
```java
import java.io.File;

/**
 *采用Google GSON来处理JSON
  * @author Tom
  *
  */
public class GsonTest {
    public static void main(String[] args) {
        testJsonObject();
        System.out.println("==========华丽丽的分割线==============");
        testJsonFile();
    }
```

代码(9) GsonTest.java



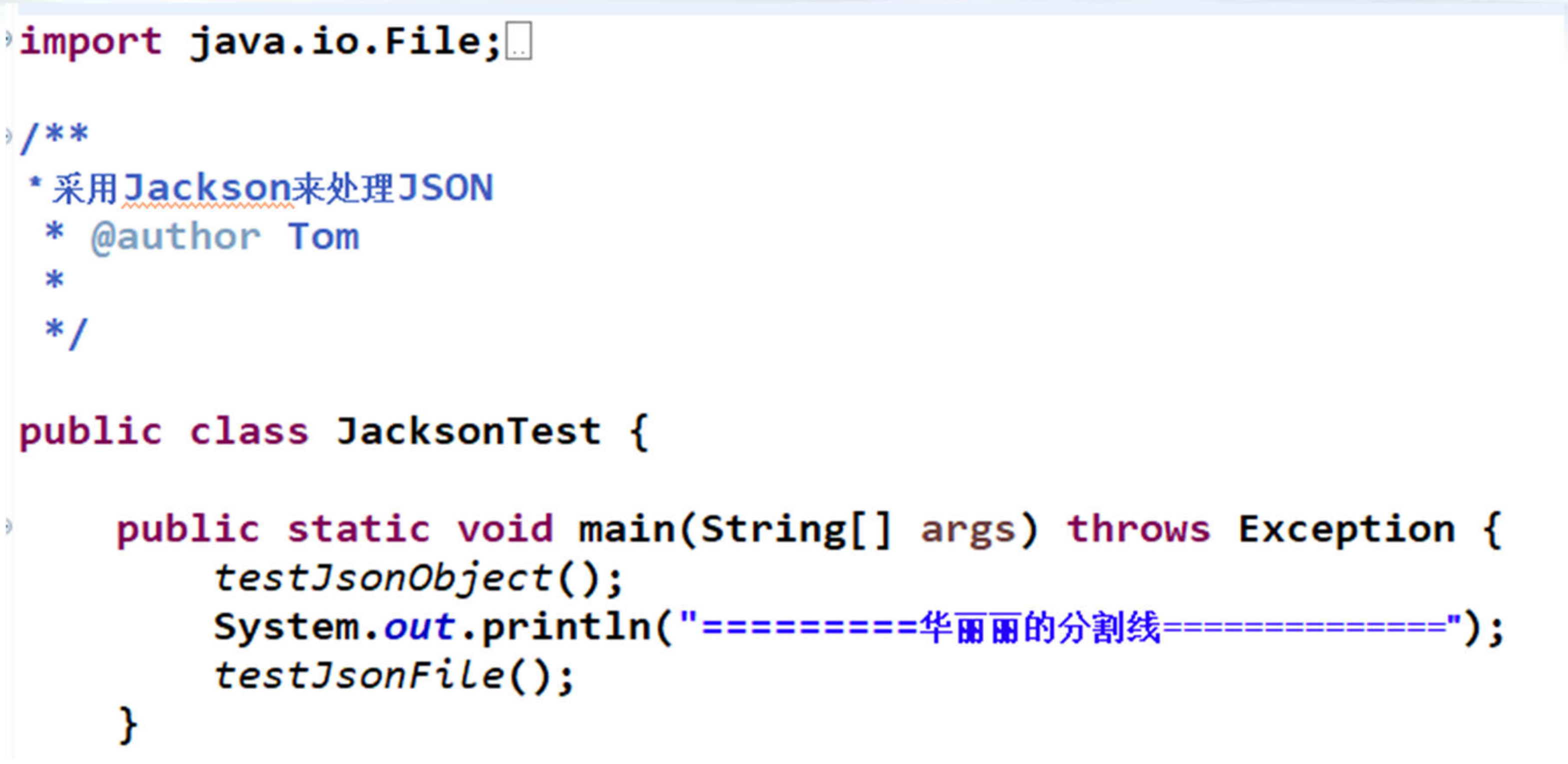
```
public static void testJsonObject() {
    //构造对象
    Person p = new Person();
    p.setName("Tom");
    p.setAge(20);
    p.setScores(Arrays.asList(60,70,80));

    //从Java对象到JSON字符串
    Gson gson = new Gson();
    String s = gson.toJson(p);
    System.out.println(s); //{"name":"Tom","age":20,"scores":[60,70,80]}

    //从JSON字符串到Java对象
    Person p2 = gson.fromJson(s, Person.class);
    System.out.println(p2.getName());  //Tom
    System.out.println(p2.getAge());   //20
    System.out.println(p2.getScores());//[60, 70, 80]

    //调用GSON的JsonObject
    JsonObject json = gson.toJsonTree(p).getAsJsonObject(); //将整个json解析为一颗树
    System.out.println(json.get("name"));  //"Tom"
    System.out.println(json.get("age"));   //20
    System.out.println(json.get("scores"));//[60,70,80]

}
```

代码(10) GsonTest.java



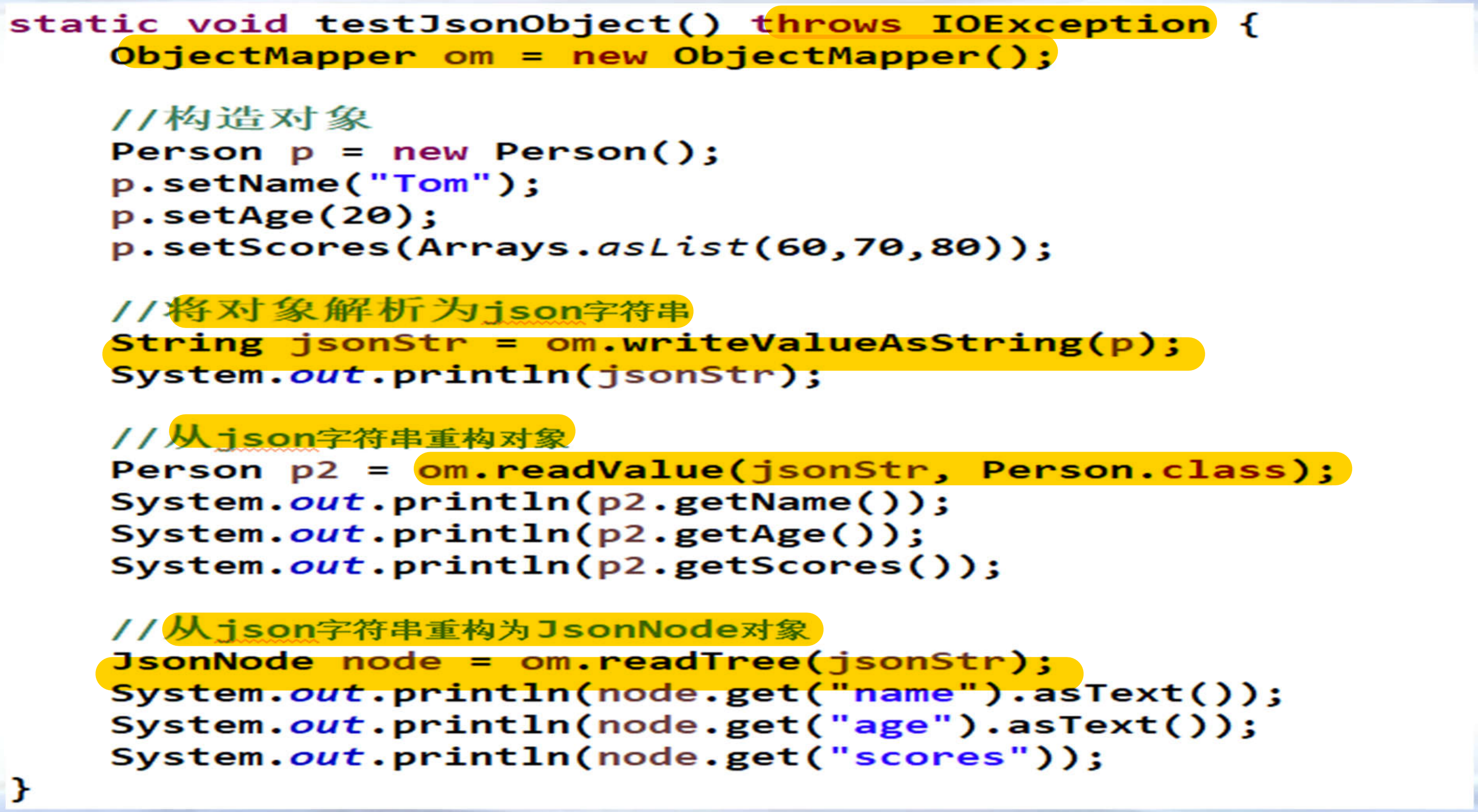
```
  public static void testJsonFile() {
    Gson gson = new Gson();
    File file = new File("books2.json");

    try (FileReader reader = new FileReader(file)) {
      List<Book> books = gson.fromJson(reader, new TypeToken<List<Book>>(){}.getType());

      for(Book book : books)
      {
        System.out.println(book.getAuthor() + ",  " + book.getTitle());
      }
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
}
```

代码(11) JacksonTest.java



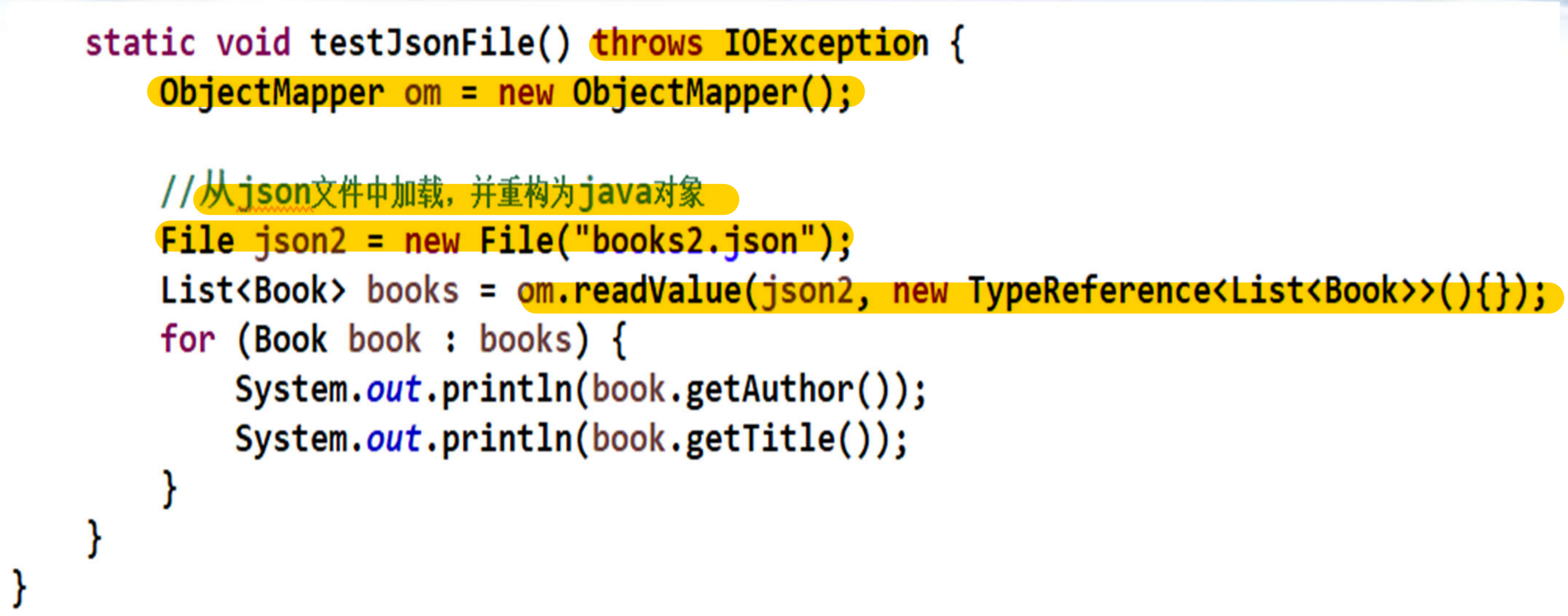
```
import java.io.File;

/**
 * 采用Jackson来处理JSON
 * @author Tom
 *
 */

public class JacksonTest {

    public static void main(String[] args) throws Exception {
        testJsonObject();
        System.out.println("=========华丽丽的分割线=============");
        testJsonFile();
    }
```

代码(12) JacksonTest.java



```java
static void testJsonObject() throws IOException {
     ObjectMapper om = new ObjectMapper();

     //构造对象
     Person p = new Person();
     p.setName("Tom");
     p.setAge(20);
     p.setScores(Arrays.asList(60,70,80));

     //将对象解析为json字符串
     String jsonStr = om.writeValueAsString(p);
     System.out.println(jsonStr);

     //从json字符串重构对象
     Person p2 = om.readValue(jsonStr, Person.class);
     System.out.println(p2.getName());
     System.out.println(p2.getAge());
     System.out.println(p2.getScores());

     //从json字符串重构为JsonNode对象
     JsonNode node = om.readTree(jsonStr);
     System.out.println(node.get("name").asText());
     System.out.println(node.get("age").asText());
     System.out.println(node.get("scores"));
}
```

代码(13) JacksonTest.java



```java
    static void testJsonFile() throws IOException {
        ObjectMapper om = new ObjectMapper();

        //从json文件中加载，并重构为java对象
        File json2 = new File("books2.json");
        List<Book> books = om.readValue(json2, new TypeReference<List<Book>>(){});
        for (Book book : books) {
            System.out.println(book.getAuthor());
            System.out.println(book.getTitle());
        }
    }
}
```

谢 谢！