Java Specialist Master Course Prep

Dr Heinz M. Kabutz

Javaspecialists.eu
java training

Welcome!

- The Java Specialist Master Course (also called *Extreme Java*) was written to address a specific need amongst developers:
  - There are lots of beginners and foundational Java courses that will teach you the basics of Java, but not the more advanced topics
  - The topics were chosen for a specific customer's needs, but they are sufficiently important to offer as a general public course
- However, this is *not* a course for programmers who are just starting with Java or who have become rusty from not working with it day-to-day
- The requirements are that you have actively programmed Java for at least 2 years. In this prep, you can hopefully discover if you are ready for this course!

Programming Ability in Java

- You will gain the most from this course if you are fairly fluent in Java. Let's try this out now:
  - Write a class called LeftRightInsert, which contains one field LinkedList of objects. Each time you insert, you swap between inserting at the beginning and end of the collection.
  - The toString() method should return the objects in the order they are stored internally
  - When you iterate, try to return the objects in the order they were inserted
  - (Optional) Make the class generics enabled
- The purpose of this exercise is for you to assess whether you need to freshen up your skills for this class. I would expect you to easily solve this kind of problem.

Given Up Yet?

- **Here is a unit test that you can run against your code**

```
import junit.framework.TestCase;
import java.util.Iterator;
public class LeftRightInsertTest extends TestCase {
  public void test() {
    for (int i = 0; i < 6; i++) {
      LeftRightInsert<Integer> lri =
          new LeftRightInsert<Integer>();
      for (int j = 0; j < i; j++) lri.add(j);
      Iterator<Integer> it = lri.iterator();
      for (int j = 0; j < i; j++) {
        assertEquals(j, it.next().intValue());
      }
    }
  }
}
```

How Long Should You Take?

- **The complete exercise, including unit testing, took me about 15 minutes**
- **I would expect students to be able to solve this within about 30 minutes, excluding the unit testing**

Partial Model Solution of iterate()

```
public Iterator<T> iterator() {
   return new Iterator<T>() {
     int size = size(); int elements = 0;
     int position = (size-1)/2; boolean left = true;
     public boolean hasNext() { return elements < size; }
     public T next() {
       T t = objects.get(position);
       elements++;
       if (left) {
         position += elements;
       } else {
         position -= elements;
       }
       left = !left;
       return t;
     } // remove() left out
```

Prior Knowledge

- **Unlike our Java Foundation Course, this Java Specialist Master Course is written for programmers with a certain prior knowledge of Java**
- **In the next few slides, I will look at each section and outline what is assumed knowledge**

Multi-Threading

- **Multi-Threading**
  - **Basics of Threads**
  - **The Secrets of Concurrency**
  - **Applied Threading Techniques**
  - **Threading Problems**
- **In this chapter, we assume you have already been exposed to the basics of threading (how to create a thread) plus very basic inter-thread communication**
  - **We do cover all these topics in this chapter, but more as a refresher than brand-new concepts**

Java IO

- **Java IO**
  - **Object Streams and Serialization**
  - **Java New IO Package**
  - **Character Streams Encoding**
- **We expect you to have some basic idea of how to use the input and output streams in Java**
  - **Print your name 20 times to a file using the following:**
    - **PrintStream**
    - **BufferedOutputStream**
    - **FileOutputStream**
  - **(Optional) write it to an ObjectOutputStream instead**

Java Memory

- **Java Memory**
  - **Garbage Collection**
  - **Tuning JVM**
  - **References**
- **We do not assume much prior knowledge for this chapter, only perhaps what a pointer in Java is**

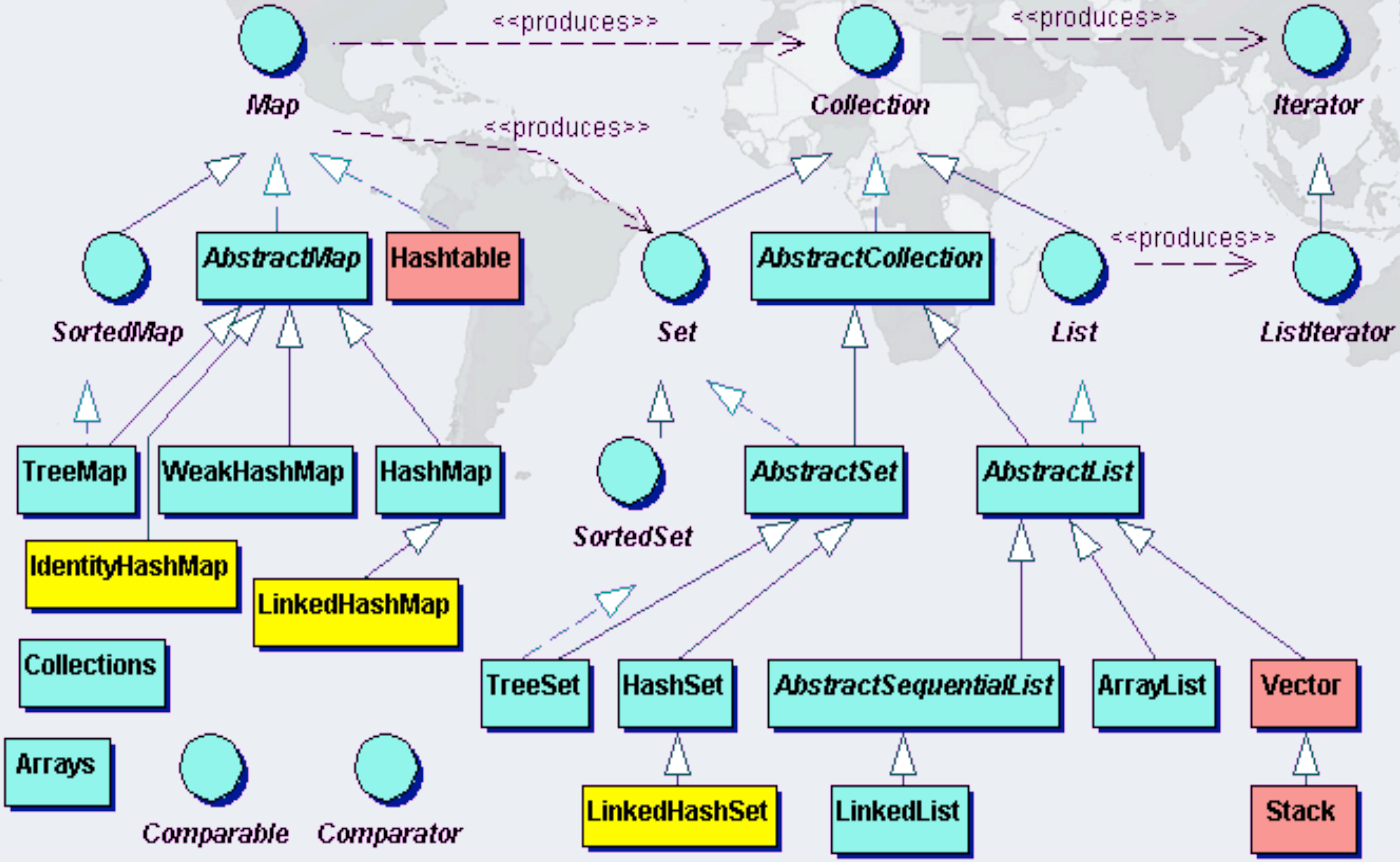
Reflection API

- **Reflection API**
  - **Overview**
  - **Manipulating Objects**
  - **Arrays**
  - **Generics**
  - **Dynamic Proxy Classes**
  - **Classes**
- **We do not assume prior knowledge for this chapter**

Data Structures

- Data Structures
  - Basics
  - Iteration
  - Sorting
  - Generics
  - Other Structures
- We assume that you are fairly familiar with the basic Java collections from JDK 1.4, though we will look at some of them again

Overview of Java 1.4 Classes



Exceptions

- **Exceptions**
  - **Error Handling**
  - **Critical Errors Inside the JVM**
  - **Best Practices**
  - **Assertions**
  - **Performance Issues**
- **We expect you to know how to declare and throw exceptions**

Java Optimizations

- **Java Optimizations**
  - **Low Level**
  - **Other Optimizations**
  - **Problem Areas**
  - **Tools**
- **We do not assume prior knowledge for this chapter**

Dates and Time Zones

- Dates and Time Zones
  - Computer Time
  - Time in Java
  - Best practices
- We do not assume prior knowledge for this chapter

Logging

- **Logging**
  - **Overview**
  - **Log4j API**
  - **Best Practices**
  - **Impact on Performance**
- **We expect you to have had some exposure to the issues involving logging**

Ready To Face The *Extreme Java* Course?

- **We hope you have not been discouraged from joining, but that you will carefully review this prep and arrive well prepared for the course**
- **The more you read up on these topics beforehand, the better your questions will be during the course**
- **My job as a trainer is to enforce what you have learned, so please arrive prepared!**

Java Specialist Master Course Prep

Dr Heinz M. Kabutz

Please contact me on heinz@javaspecialists.eu if you have any questions