

Quiz Yourself: Read and Set the Locale (Advanced)

QUIZ

Quiz Yourself: Read and Set the Locale (Advanced)

For happy users, specify the locale correctly.

by *Simon Roberts and Mikalai Zaikin*

September 27, 2019

If you have worked on our quiz questions in the past, you know none of them is easy. They model the difficult questions from certification examinations. The levels marked “intermediate” and “advanced” refer to the exams, rather than the questions. Although in almost all cases, “advanced” questions will be harder. We write questions for the certification exams, and we intend that the same rules apply: Take words at their face value and trust that the questions are not intended to deceive you, but straightforwardly test your knowledge of the ins and outs of the language.

The objective of this question is to read and set the locale by using the `Locale` object.

Which of the following is true? Choose one.

- A. A `Locale` can represent both a country and a language.
- B. When creating a `Locale`, a country must be specified.
- C. A language is represented by exactly two characters.
- D. It's always possible to use a predefined `static final` constant from the `Locale` class, rather than create a new instance of a `Locale`.

Answer. The class `java.util.Locale` represents aspects of geographical, language, political, and cultural preferences. Not all of these aspects are required. The idea is to allow the software to provide the best “fit” it can for a given user's needs and preferences.

For example, most users would think it an absolute requirement that a user interface be presented in their first language or at least in a language they understand. However, they might be willing to use the software even if it presents the day, month, and year elements of dates in an order that's not their country's customary format.

Based on that, it's perhaps not a surprise that the elements of a `Locale` object in Java are prioritized, with language taking highest priority and region being assigned a lower priority.

The `Locale` class has three constructors. One requires only a language, one requires a language and country, and the third requires both these elements and the `variant` element. These are the constructors:

- `Locale(String language)`
- `Locale(String language, String country)`
- `Locale(String language, String country, String variant)`

Two key points should be noted from the discussion so far:

First, `Locales` can represent both language and country (and more). This makes option A correct.

Second, although language is the highest priority element and cannot be omitted from any of the constructors, the geographic region is secondary and can be omitted. Understanding the nature of `Locales`, perhaps supported by knowledge of the API features, tells you that option B is incorrect.

The representation of a language is commonly done using a two-character lowercase string; however, the standards (technically IETF BCP 47 and ISO 639) and the API documentation note that three characters are allowed. Three-character tags are rare, but the API documentation notes `kok` (Konkani) as an example. Further, a subtag might extend this to eight characters. All this tells you that option C is incorrect.

Konkani is a language spoken in India by approximately 7 million people.

It's worth noting that this element of the question falls into the rather doubtful category of rote knowledge. We and the authors of the exam dislike such questions. Frankly, it's pretty hard to write a question about this particular topic that doesn't involve something merely factual. But, notice that if you have a reasonable working understanding of the nature and use of a `Locale`, you could immediately select option A with complete confidence and never worry about the others. Such opportunities abound in many situations that appear to demand knowledge of dry facts.

The release of Java 8 brought with it acknowledgment that constructors are not generally a very good thing to expose as part of a public API. They're necessary for the construction of an object, and non-private ones will be needed if you want to use class inheritance. However, as part of a public API, they're generally bad for several reasons.

One of those reasons is that if you invoke `new`, you either get a new object or you get an exception. This means that handing out objects from a pool is not possible. Now, pooling *mutable* objects in Java (or any reference-based language) is almost universally unwise, but pooling immutable objects that are taken from a modest range of possible values (think "runtime extensible enum") makes sense and is an increasingly common API feature.

One way to minimize the creation of unnecessary duplicates of immutable objects is to provide predefined constants, which the `Locale` class does for about eight countries in a number of variations. However, that's not even close to all the countries that are covered, and it's far from all the combinations. Therefore, option D is incorrect.

The correct answer is option A.



Simon Roberts

Simon Roberts joined Sun Microsystems in time to teach Sun's first Java classes in the UK. He created the Sun Certified Java Programmer and Sun Certified Java Developer exams. He wrote several Java certification guides and is currently a freelance educator who publishes recorded and live video training through Pearson InformIT (available direct and through the O'Reilly Safari Books Online service). He remains involved with Oracle's Java certification projects.

Mikalai Zaikin



Mikalai Zaikin is a lead Java developer at IBA IT Park in Minsk, Belarus. During his career, he has helped Oracle with development of Java certification exams, and he has been a technical reviewer of several Java certification books, including three editions of the famous *Sun Certified Programmer for Java* study guides by Kathy Sierra and Bert Bates.

Share this Page



Contact

US Sales: +1.800.633.0738
Global Contacts
Support Directory
Subscribe to Emails

About Us

Careers
Communities
Company Information
Social Responsibility Emails

Downloads and Trials

Java for Developers
Java Runtime Download
Software Downloads
Try Oracle Cloud

News and Events

Acquisitions
Blogs
Events
Newsroom

