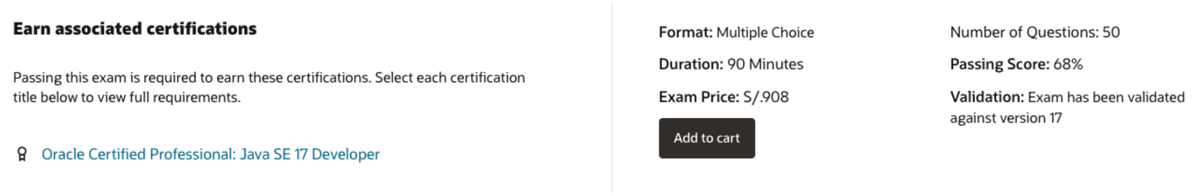
Actualizado el 17 de Dic 2022

Java SE 17 Programmer I | 1Z0-829

Nro. Horas: 48

Este es un taller de preparación para la certificación Java SE 17 Developer de Oracle. Este examen tiene el código IZ0-829, costa de **50 preguntas**, una duración de **90 minutos** y se necesita mínimo un **68% de aprobación**. Es de múltiple elección.



Earn associated certifications

Passing this exam is required to earn these certifications. Select each certification title below to view full requirements.

Oracle Certified Professional: Java SE 17 Developer

Format: Multiple Choice

Duration: 90 Minutes

Exam Price: S/.908

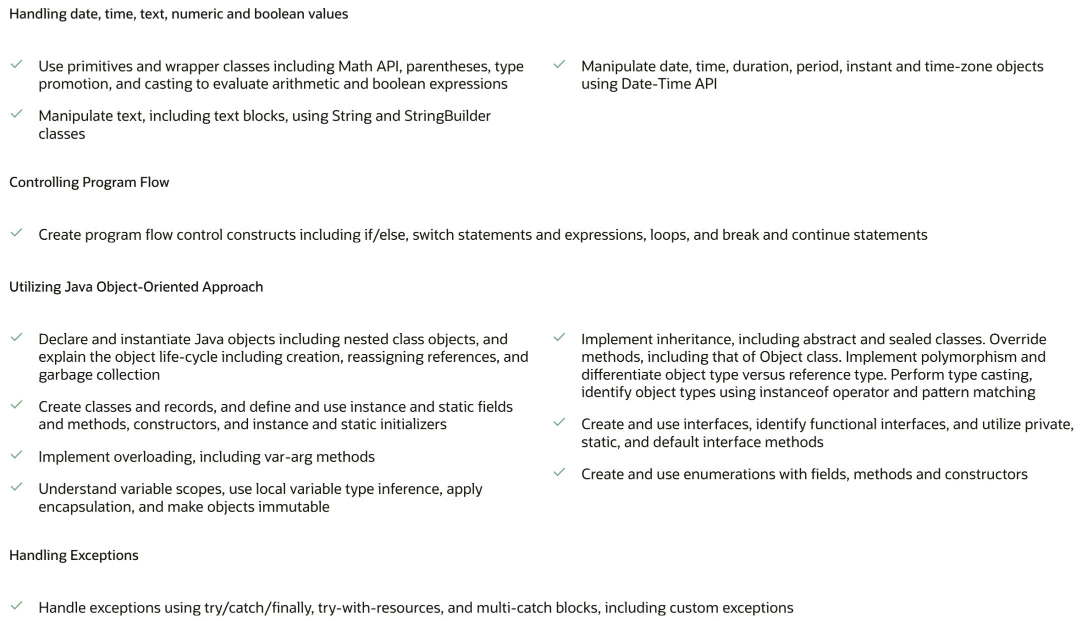
Number of Questions: 50

Passing Score: 68%

Validation: Exam has been validated against version 17

TOPICOS

Esos son los tópicos que se estudiaran en el taller:



Handling date, time, text, numeric and boolean values

- ✓ Use primitives and wrapper classes including Math API, parentheses, type promotion, and casting to evaluate arithmetic and boolean expressions
- ✓ Manipulate text, including text blocks, using String and StringBuilder classes
- ✓ Manipulate date, time, duration, period, instant and time-zone objects using Date-Time API

Controlling Program Flow

- ✓ Create program flow control constructs including if/else, switch statements and expressions, loops, and break and continue statements

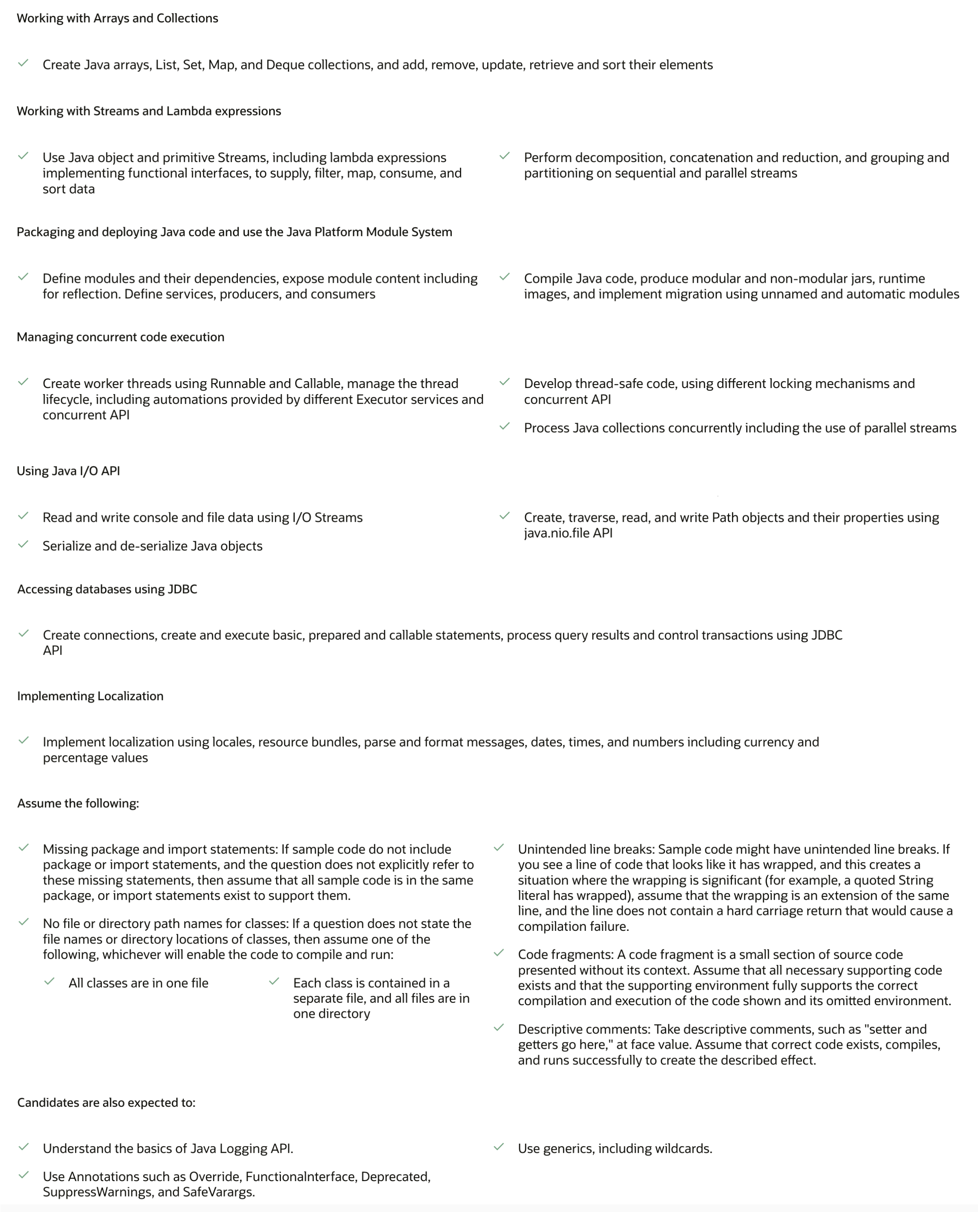
Utilizing Java Object-Oriented Approach

- ✓ Declare and instantiate Java objects including nested class objects, and explain the object life-cycle including creation, reassigning references, and garbage collection
- ✓ Create classes and records, and define and use instance and static fields and methods, constructors, and instance and static initializers
- ✓ Implement overloading, including var-arg methods
- ✓ Understand variable scopes, use local variable type inference, apply encapsulation, and make objects immutable
- ✓ Implement inheritance, including abstract and sealed classes. Override methods, including that of Object class. Implement polymorphism and differentiate object type versus reference type. Perform type casting, identify object types using instanceof operator and pattern matching
- ✓ Create and use interfaces, identify functional interfaces, and utilize private, static, and default interface methods
- ✓ Create and use enumerations with fields, methods and constructors

Handling Exceptions

- ✓ Handle exceptions using try/catch/finally, try-with-resources, and multi-catch blocks, including custom exceptions

Actualizado el 15 Ago 2021



Working with Arrays and Collections

- ✓ Create Java arrays, List, Set, Map, and Deque collections, and add, remove, update, retrieve and sort their elements

Working with Streams and Lambda expressions

- ✓ Use Java object and primitive Streams, including lambda expressions implementing functional interfaces, to supply, filter, map, consume, and sort data
- ✓ Perform decomposition, concatenation and reduction, and grouping and partitioning on sequential and parallel streams

Packaging and deploying Java code and use the Java Platform Module System

- ✓ Define modules and their dependencies, expose module content including for reflection. Define services, producers, and consumers
- ✓ Compile Java code, produce modular and non-modular jars, runtime images, and implement migration using unnamed and automatic modules

Managing concurrent code execution

- ✓ Create worker threads using Runnable and Callable, manage the thread lifecycle, including automations provided by different Executor services and concurrent API
- ✓ Develop thread-safe code, using different locking mechanisms and concurrent API
- ✓ Process Java collections concurrently including the use of parallel streams

Using Java I/O API

- ✓ Read and write console and file data using I/O Streams
- ✓ Serialize and de-serialize Java objects
- ✓ Create, traverse, read, and write Path objects and their properties using java.nio.file API

Accessing databases using JDBC

- ✓ Create connections, create and execute basic, prepared and callable statements, process query results and control transactions using JDBC API

Implementing Localization

- ✓ Implement localization using locales, resource bundles, parse and format messages, dates, times, and numbers including currency and percentage values

Assume the following:

- ✓ Missing package and import statements: If sample code do not include package or import statements, and the question does not explicitly refer to these missing statements, then assume that all sample code is in the same package, or import statements exist to support them.
- ✓ No file or directory path names for classes: If a question does not state the file names or directory locations of classes, then assume one of the following, whichever will enable the code to compile and run:
  - ✓ All classes are in one file
  - ✓ Each class is contained in a separate file, and all files are in one directory
- ✓ Unintended line breaks: Sample code might have unintended line breaks. If you see a line of code that looks like it has wrapped, and this creates a situation where the wrapping is significant (for example, a quoted String literal has wrapped), assume that the wrapping is an extension of the same line, and the line does not contain a hard carriage return that would cause a compilation failure.
- ✓ Code fragments: A code fragment is a small section of source code presented without its context. Assume that all necessary supporting code exists and that the supporting environment fully supports the correct compilation and execution of the code shown and its omitted environment.
- ✓ Descriptive comments: Take descriptive comments, such as "setter and getters go here," at face value. Assume that correct code exists, compiles, and runs successfully to create the described effect.

Candidates are also expected to:

- ✓ Understand the basics of Java Logging API.
- ✓ Use Annotations such as Override, FunctionalInterface, Deprecated, SuppressWarnings, and SafeVarargs.
- ✓ Use generics, including wildcards.

Prerrequisitos:

- Conocimientos de Java básico.

Actualizado el 15 Ago 2021

Formas de Pago:

Transferencia Bancaria:

- BCP 191-30759925-0-29 (Ahorro Soles)
- BBVA 0011-0339-0200168694 (Ahorro Soles)
- INTERBANK 200-3116727850 (Ahorro Soles)
- SCOTIABANK 174-0055213 (Ahorro Soles)

PayPal:

- Enviar el pago a jamdiazdiaz@gmail.com

Pago con Tarjeta de Débito y Crédito

- Solicitar el pago con tarjeta de débito y crédito al + 51 939 965 148 (niubiz)

Consultas:

Si tiene alguna consulta o comentario nos puede escribir al WhatsApp al +51939965148 o +51966025115