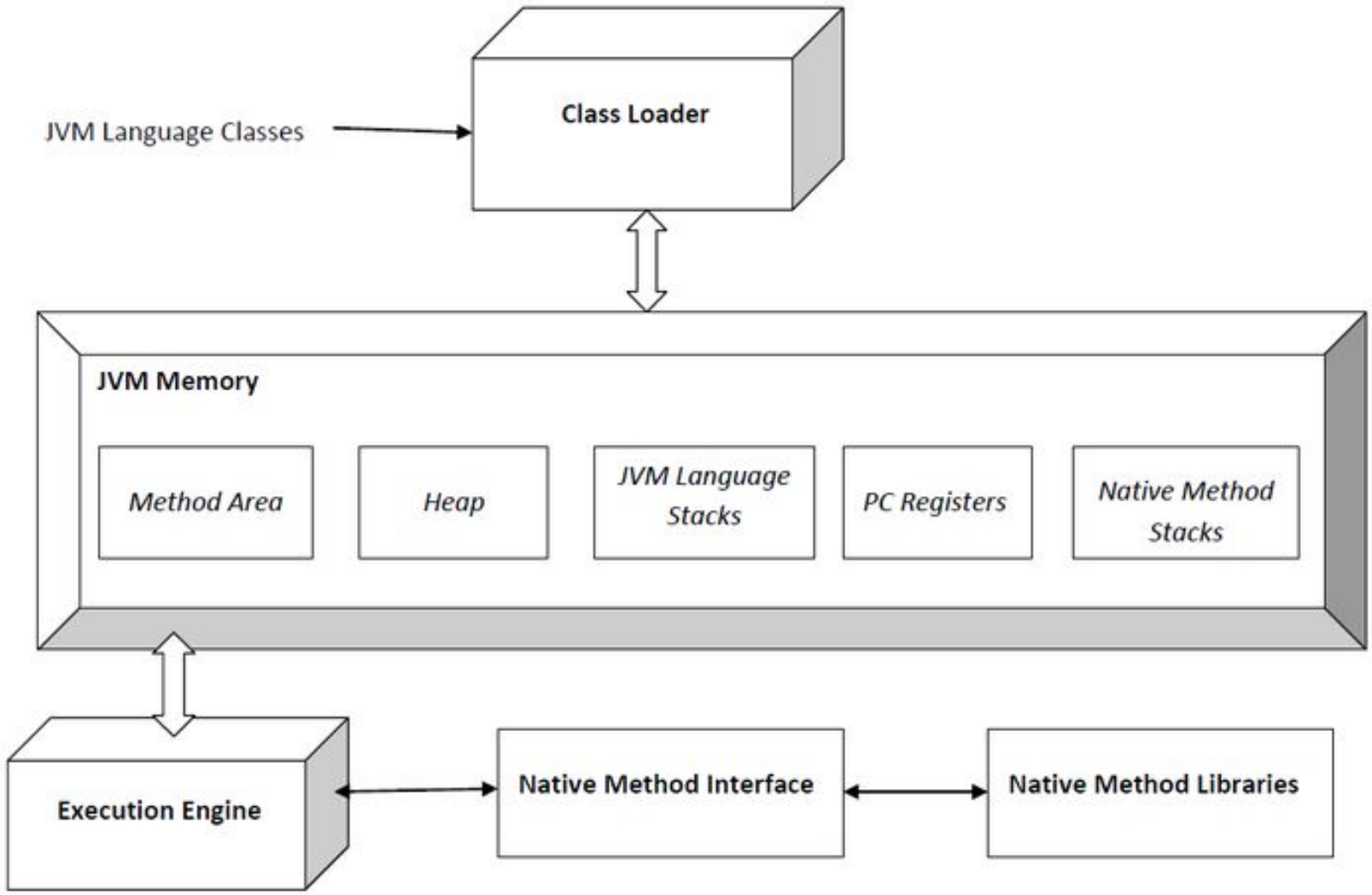
YAYASAN PENERAJU - GKK CONSULTANTS

Java JW C4 & C5 2024 - Java Professional

AYHAM AJAJ

0. Introduction

How Java Works



Introduction to Java Foundations Exam

1. The Java Foundations Exam exam (1Z0-811) covers the fundamentals of Java SE 8 programming, such as the structure of classes and interfaces, variables of different data types, methods, operators, arrays, decision constructs, and loops.
2. The exam includes handling exceptions and a few commonly used classes from the Java API like String, StringBuilder, and ArrayList.
3. This exam doesn't include a lot of Java 8-specific language features.
4. It includes an introduction to functional-style programming with lambda expressions. It partially covers the new Date and Time API.
5. It certifies that an individual possesses a strong foundation in the Java programming language.

Java Basics

1. Define the scope of variables
2. Define the structure of a Java class
3. Create executable Java applications with a main method; run a Java program from the command line, including console output
4. Import other Java packages to make them accessible in your code
5. Compare and contrast the features and components of Java, such as platform independence, object orientation, encapsulation, and so on

Working with Java data types

1. Declare and initialize variables (including casting of primitive data types)
2. Differentiate between object reference variables and primitive variables
3. Know how to read and write to object fields
4. Explain an object's lifecycle (creation, "dereference by reassignment," and garbage collection)
5. Develop code that uses wrapper classes such as Boolean, Double, and Integer

Using Operators and decision constructs

1. Use Java operators, including parentheses to override operator precedence
2. Test equality between Strings and other objects using == and equals ()
3. Create if and if/else and ternary constructs
4. Use a switch statement

Creating and using arrays

1. Declare, instantiate, initialize, and use a one-dimensional array
2. Declare, instantiate, initialize, and use a multidimensional array

Using loop constructs

1. Create and use while loops
2. Create and use for loops, including the enhanced for loop
3. Create and use do-while loops
4. Compare loop constructs
5. Use break and continue

Working with methods and encapsulation

1. Create methods with arguments and return values, including overloaded methods
2. Apply the static keyword to methods and fields
3. Create and overload constructors, including impact on default constructors
4. Apply access modifiers
5. Apply encapsulation principles to a class
6. Determine the effect on object references and primitive values when they are passed into methods that change the values

Working with inheritance

1. Describe inheritance and its benefits
2. Develop code that demonstrates the use of polymorphism, including overriding and object type versus reference type
3. Determine when casting is necessary
4. Use super and this to access objects and constructors
5. Use abstract classes and interfaces

Handling exceptions

1. Differentiate among checked exceptions, unchecked exceptions, and errors
2. Create a try-catch block and determine how exceptions alter normal program flow
3. Describe the advantages of exception handling
4. Create and invoke a method that throws an exception
5. Recognize common exception classes (such as `NullPointerException`, `ArithmeticException`, `ArrayIndexOutOfBoundsException`, `ClassCastException`)

Working with selected classes from the Java API

1. Manipulate data using the StringBuilder class and its methods
2. Creating and manipulating Strings
3. Create and manipulate calendar data using classes from `java.time.Local-DateTime, java.time.LocalDate, java.time.LocalTime, java.time.format.DateTimeFormatter, and java.time.Period`
4. Declare and use an ArrayList of a given type
5. Write a simple lambda expression that consumes a lambda predicate expression