Java Interview questions notes

History of Java

Java is an object-oriented programming language released by Microsystems in 1995.

Write Once, Run Anywhere(WORA), cross platforms unlike c++.

Borrowed syntax from c++, but provide automatic memory management and eliminate multiple inheritance.

The java virtual machine

WORA is possible because of JVM, in java, code is compiled into a virtual machine language called bytecode. The JVM acts as an intermediary between bytecode and the physical machine.

Every platform that supports Java has its own implementation of the JVM. Java applications are portable because every JVM adheres to a standard interface. The distribution package of   the JVM and standard libraries is called the Java Runtime Environment (JRE). The distribution package of the JRE and development tools, such as the compiler and debugger, is called the JDK.

Procedural Programming vs Object-Oriented Programming

Procedural programming is writing code that executes a series of linear procedures to produce a result. Object-oriented programming is writing code that uses objects to encapsulate attribute and behavior. Procedural code is easier to use in small projects or in multithreaded environments due to its stateless nature, but object-oriented code is far more flexible and easier to maintain.

What is the WORA principle? Why is it beneficial?

Write once and run anywhere, due to the java virtual machine, java code could run on different platforms, it could reduce programmer’s workload, which means they do not need to implement the same application on different platforms, it is time-consuming.

How could java applications run on multiple platforms?

Due to the java virtual machine, every platform has its own implementation of JVM, java code is compiled in an intermediary language called bytecode. JVM translate bytecode into machine language and then run the code.

What is the difference between the JRE and the JDK?

JRE is a package of JVM and java standard libraries.

JDK is a package of JRE and development tools such as debugger, compiler.

What is the difference between procedural programming and object-oriented programming?

Procedural programming is writing linear procedures to produce a result. It is suitable in small projects or in multithreaded environments due to its stateless nature.

Object-oriented programming is writing objects that encapsulate attributes and behaviors. It is flexible to use and easier to maintain.

Object-Oriented Concepts

What is the purpose of abstraction in software development?

Abstraction is the act of perceiving an entity from a narrow perspective. The goal of abstraction is to reduce the complexity in software systems.

What is encapsulation? How does java support it?

Encapsulation is a technique that encourages abstraction by hiding information. Java encourage encapsulation through the use of interfaces and by providing access modifiers that could limit the visibility of classes, fields and methods.

What is polymorphism? How does java support it?

Polymorphism is a technique that encourages abstraction by allowing an entity to assume multiple forms. In Java, an object can take on the form of any parent in its hierarchy or any interface in its hierarchy.

Class&Object

What is the difference between a class and an object?

Class is the blueprint of object, object is an instance of class.

Class define object’s fields and methods, object use new keyword to initialize class.

When initialize an object, class loader will load the class file and all of its superclasses first, after the one-time initialization of the class, the object will invoke a special method called a constructor method. Superclass first, then subclasses.

What happens when an object is instantiated for the first time?

Java virtual machine will load its class and all its superclasses into the memory. After the initialization of the class, the object will invoke a special method called constructor to initialize itself. The constructor method will recursively invoke all of the constructors of its superclasses.

What is the difference between a primitive type and an object?

A primitive type is a fixed-size data type that is predefined and reserved as a keyword.

Primitive types are not objects, but every primitive type has a corresponding wrapper object.

What is the difference between autoboxing and unboxing?

Autoboxing means primitive types convert to its wrapper objects

Unboxing mean wrapper objects convert to its primitive types.

Array

What is an array?

Array is an object that hold a fixed number of values of a single type.

Array could cast to its superclass type. String[] 🡪 Object[]

Strings

How is a String different from a regular object?

Strings are unique objects that are used to represent text.

Can be initialized without new keyword.

Can be concatenated via the overload + operator.

Cache it in a process called interning.

What is the difference between a StringBuilder and a StringBuffer?

StringBuffer is thread-safe, but StringBuilder is not.

If your program is single thread, use StringBuilder is more efficient; if it is a multiple threads, use StringBuffer ensure its thread safe.

Why are enums superior to String or Integer constants?

Represent a set of single-instance constants.

Could contain methods, implement interfaces and provide type safety.

Enums could not be subclassed or extend any class besides the implicitly extended Enum class.

What is difference between package-by-layer and package-by-feature?

Package-by-feature is preferable.

What is the difference between a method declaration and a method signature?

Method declaration contains a list of modifiers, a return type, a method name, a list of parameter types and their corresponding names and a list of throwable exceptions.

Method signature contains method name, the types and order of its parameters, not parameters’ names included.

What is recursive method?

A method call itself, a alternative to loop.

What is the final keyword used for?

Final keyword indicates whether a reference can be changed after being assigned.

Final class could not be extended. All its methods are final methods.

Final methods could not be overriding.

Final fields could not be changed, they are constants.

What is the static keyword used for?

Static keyword indicates a property belongs to a class or an object instance.

Why can’t a static method access a nonstatic field?

Since static method usually call by the class name, nonstatic fields belongs to objects. When calling the method, there is no object, so these nonstatic fields are not initialized.

What are access modifiers used for? What are the different types?

Indicate the visibility of classes, fields and methods.

Public: visible to all classes

Protected: visible to subclasses and classes in the same package

Default: visible to the same package

Private: visible to the inner class

What are annotations used for?

Annotations are applied to fields, methods, classes and packages used to embed metadata alongside code.

The Object Superclass

Every class in java is a directly or indirectly subclass of the Object class

The clone() method was originally designed to return a copy of an object that implements the Cloneable interface. A clone can either be a shallow copy, which shares the same references as the original object, or a deep copy, which copies the values of the original object into new objects.

The equals() method compares two objects for equality. The default implementation relies on   the identity operator (= =) to determine whether two objects point to the same address in   memory. Subclasses are encouraged to override this method to test whether two objects contain the same information rather than the same location in memory. Note that if you override the equals() method you must by contract override the hashCode() method as well.

The hashCode() method digests the state of an object into an integer, which is primarily useful for hash table data structures. By default, the hash code is implemented by converting the internal address of an object into an integer. The hash code must be consistently returned and should always return equal values for objects that are equal according to the equals() method.

The toString() method returns a textual representation of an object, which is primarily useful for logging and debugging. By default, the toString() method returns the class of the object followed by a hexadecimal representation of its hash code value.

The getClass() method returns a Class object that contains information about a class and utility methods for reflection-based access to fields and methods. The getClass() method is final and cannot be overridden by subclasses.

The finalize() method was originally designed to be invoked before an object was destroyed by the garbage collector. However, an object might not become eligible for garbage collection if it’s never dereferenced or if the application exits before the garbage collector runs. It’s generally discouraged to rely on this method for cleanup operations due to its uncertainty and the possibility that an object can be unintentionally revived by creating additional references to it.

The final methods wait(), notify(), and notifyAll() provide low-level concurrency operations that allow communication between threads. For example, one thread could halt its execution until it receives a notification from another thread. Java provides high-level concurrent data structures in the java.util.concurrent package.

What is the difference between a shallow copy and a deep copy?

A shallow copy just share the original one’s address in the memory, deep copy means copy all values of the original one to a new object.

Why is a copy constructor preferable to the clone method?

If I use this default method, I do not know whether it is a shallow or deep one. And copy constructor offers more flexibility and a cleaner contract.

What is the difference between the identity operator and the equals() method?

Identity operator compares the address in the memory, equals() by default also compare addresses of two object in the address. But we could override it to fit our requirements.

What is the relationship between the hashCode() method and the equals() method?

hashCode() method convert the internal address of an object into an integer. The method must be returned and should always return equals values for objects that are equal according to the equals() method.

What is the default implementation of the toString() method?

By default, the toString() method will return its hash code value of the object in hexdecimal representation.

Why is the finalize() method unreliable for cleanup operations?

Because an object may never dereference or if the application exits before before the garbage collector runs.

Composition & Inheritance

What is the difference between composition and inheritance?

Composition means having an instance of another class as a field of a class, inheritance indicates a class extends another class.

What is the difference between method overriding and method overloading?

Method overriding allows a subclass to change the functionality of a superclass. This occurs when a subclass defines a non-static method with the same method signature as a parent method. Method overloading occurs when two or more methods in a class have the same name with different method signatures.

Abstract Classes & Interfaces

Abstract classes are designed to be extended and cannot be instantiated.

Interfaces promote polymorphism by describing a form that an object can take.

What is the difference between an abstract class and an interface?

Abstract classes are used to implement an inheritance hierarchy and simplify the work of subclasses. Interfaces are used to introduce a layer of abstraction and decouple modules of codes.

Abstract class use extends, a class could only extend one class; interfaces use implements, a class could implement multiple interfaces.

Abstract classes could have its own fields, concrete methods and abstract methods; interfaces could just have constant fields and abstract methods.

How would you determine whether to use an abstract class or an interface?

If I want to represent an inheritance relationship and subclasses may have its different implementations for some methods of the superclass, I would choose the abstract class; if I want different objects have its own implementations of their behaviors, I would choose interface.

Why cannot a class be declared both final and abstract?

An abstract class could not be instantiated and must be extends, a final class could not be extended.

What is the value of designing a codebase around the use of interfaces?

Each type of class could have its own implementations of its functionality.

What are anonymous classes used for?

Reduce the verbosity of the language.

A closure is a block of code that can be passed around while maintaining access to variables of the enclosing scope.

What is a lambda expression?

A lambda expression is a single-line representation of an anonymous class that contains a single method.

Exceptions

Exceptions are special objects that are thrown whenever an error interrupts the normal execution of code. All exceptions are descendents of the Throwable class and are divided into two categories: unchecked exceptions and checked exceptions.

Unchecked exceptions extend from RuntimeException class represent a defect in the application, without forcing its caller to build error-handling policy. Common unchecked exceptions include NullPointerExceptions, IllegalArgumentExceptions and ClassCastExceptions.

Checked exceptions extend from Exception class represent a defect that occurs outside of the control of the application with forcing its caller to build an error-handling policy. Callers must either surround the method in a try-catch block or add the exception to its method declaration to push the responsibility further up the call stack. Common checked exception include IOExceptions, FileNotFoundExceptions and InterruptedExceptions.

Unchecked Exceptions vs Checked Exceptions

If a client can be expected to recover from an exception, make it a checked exception. If a client cannot do anything to recover from the exception, make it an unchecked exception.

unchecked exceptions are preferred because it allows clients to choose whether an error-handling policy is appropriate.

Errors

An Error is thrown by the JVM to indicate that a fatal condition has occurred. Errors extend the Throwable class directly, which gives them the behavior of unchecked exceptions. Common errors include OutOfMemoryErrors or StackOverflowErrors.

The try keyword delineates a block of code that might throw an exception. The catch keyword designates a handler for a specific type of exception. The finally keyword delineates a block of code that will be invoked regardless of whether an exception was thrown.

What is an exception?

Exception is a special object in java that are thrown whenever an error interrupts the execution of normal code. All exceptions are subclasses from Throwable class and divided to two categories as checked exceptions and unchecked exceptions.

What is the difference between an unchecked and a checked exception?

Unchecked exceptions extend from RuntimeException class represent a defect in the application. It does not force the method caller to build error-handling policy. Common unchecked exceptions include ClassCastExceptions, NullPointerExceptions and IllegalArgumentExceptions. Checked exceptions extend from Exception class represent a defect outside the control of the application with forcing its caller to build error-handling policy. Callers must surround the method in try-catch-finally black or add the exception in its method declaration to push the responsibility further up the call stack. Common checked exception include IOExceptions, FileNotFoundExceptions and InteruptedExceptions.

How would you determine whether to use an unchecked or a checked exception?

If a client can be expected to recover from an exception, make it a checked exception. If a client cannot do anything to recover from the exception, make it an unchecked exception.

How does a try-catch-finally block work?

The try keyword delineates a block of code that might throw an exception. The catch keyword designates a handler for a specific type of exception. The finally keyword delineates a block of code that will be invoked regardless of whether an exception was thrown. It provides cleanup operations.

How does the try-with-resources statement work?

It allows resources to implement the autocloseable, then JVM will automatically call the close() method on the resources, eliminating the need for a finally block.

Generics

What is the difference between a compile-time error and a runtime error?

A compile time error happens in compiling such as syntax errors; a runtime error happens during the execution of the application such as type casting error. Compile time error is easier to handle since compiler could tell you where the problem is. But run time errors are more difficult to detect and may cause unpredictable behavior of the application.

What is the purpose of generics?

Generics were introduced to prevent runtime errors caused by invalid type casting.

What are the difference types of generic wildcards?

List<? Extends Number> allows the Number types and all subclasses of Number.

List<? Super Number> allows the Number type and all superclasses of Number.

What is type erasure?

To preserve backwards compatibility, generic types are removed by the compiler and replaced with object casts in a process called type erasure.

What are some of the limitations of generics?

Could not catch generic exceptions since the compiler has removed generic types.

Could not implement interfaces according to the generic types since the compiler has removed it and all interfaces are the same.

Concurrency

What is the lifecycle of a Thread?

A thread is born by instantiating a subclass of the Thread class or by implementing Runnable interface. A thread enters a runnable state when its start() method is invoked. The JVM uses a priority-based scheduling algorithm to determine which thread to execute. A thread enters a running state when its run() method is invoked by the scheduler. a thread can transition into a blocked state if it needs to acquire a lock or if it waits for a notification from another thread. A thread dies after the run() method completes.

Why is synchronization necessary on shared resources?

Since the operations of multiple threads can interfere.

A deadlock occurs when two threads wait for each to proceed.

What is used as a lock for synchronized static and synchronized non-static methods?

The synchronize keyword is used to designate synchronized methods and statements. A synchronized block of code is guarded by a lock that can only be acquired by one thread. In a synchronized static method, the singleton Class object is implicitly used as the lock. In a synchronized non-static method, the object instance is implicitly used as the lock.

What would happen if two different threads hit two synchronized non-static methods on the same object simultaneously?

What would happen if two different threads hit a synchronized static method and synchronized non-static method on the same object simultaneously?

What is volatile keyword guarantee about a variable?

The volatile keyword is used to indicate that the value of a variable may be modified by multiple threads.

What two things does the synchronize keyword guarantee about a block of code?

The guarantee that a   memory write will be   visible to   another block of code is called a happens-before relationship. Both the volatile keyword and the synchronize keyword guarantee a happens-before relationship, however the synchronize keyword also guarantees atomic access to a block of code.

What are some built-in concurrent data structure?

CopyOnWriteArrayList, ConcurrentHashMap, BlockingQueue

What is the executor framework?

The executor framework provides a layer of abstraction over multithreaded task execution. The ExecutorService manages a thread pool that accepts Runnable or Callable tasks. Submitting a task immediately returns a Future object, which contains methods that return the status and result of a running task. The executor framework effectively decouples tasks from their execution policies.

What is a ThreadLocal variable?

The ThreadLocal class allows values to be stored inside of the currently running Thread object.

What are atomic variables?

Variables with built-in concurrency. AtomicInteger, AtomicLong and Atomic Boolean.

Memory Management

How does the JVM divide memory on the heap?

Java objects reside in a dynamically sized area of memory called the heap. The heap is divided into young generation objects and old generation objects. The young generation divide into three parts as Eden, from\_survivor and to\_survivor space. All new objects are created on the Eden space, unless it is full and objects will be created in from\_survivor, to\_survivor space is always empty until the garbage collection happens. After many times gc in young generation, if objects are still alive, then they will be moved to old generation where garbage collection seldom happens.

What is the standard algorithm for garbage collection?

Gc is invoked when the heap fills up. The default algorithm is mark and sweep.in the first step, objects that are referenced in memory are identified and marked. In the second step, unmarked objects are deleted and freeing up chunks of memory. The remaining memory can then be compacted into contiguous blocks.

What are memory leak? How can they be identified?

Memory leak indicates an object could not be referenced anymore since the heap is full. Debugging a memory leak usually requires the help of a profiler. A profiler can analyze the heap dump created when the JVM crashes and recreate the reference hierarchy to pinpoint where most memory is being retained.

What are four types of references?

Strong reference: A strong reference is a typical reference such as assigning an object to   a variable or putting an object in an array. A strong reference to an object guarantees that the object will remain on the heap. An object is eligible for garbage collection if it cannot be accessed through any chain of strong references. Strong references can accumulate over time. When an application uses more memory than the allocated heap space, the result is an OutOfMemoryError. When an application fills up the heap with unintentional references, the result is a memory leak. OutOfMemoryErrors can be fixed by allocating a larger heap size through JVM arguments, but a memory leak could eventually fill up a heap of any size.

Soft reference: A soft reference is an object that can only be retrieved by invoking the get() method on a SoftReference container. Softly reachable objects will not be eagerly garbage collected, making them ideal for caches. For example, if we stored SoftReference < Star > values inside of our map, our cache could grow until it filled the heap, at which point the get() method would start returning null for stars that were garbage collected. We could then treat these values as if they were expired cache entries.

Weak reference: A weak reference is an object that can only be retrieved by invoking the get() method on a WeakReference container. Weakly reachable objects will be eagerly garbage collected, making them ideal for short-lived object associations. For example, if we allowed our users to edit stars, we could use a ConcurrentMap to synchronize access to a star by associating its name to a User. However, if the user closed their browser prematurely the star would remain locked in the map. Assuming that the User was also stored as a strong reference in a session object, we could use a WeakReference < User > to determine when that session expired because the get() method would return null soon after the session was garbage collected.

Phantom reference: A phantom reference is an object that is wrapped inside of a PhantomReference container. However, invoking the get() method on a PhantomReference will always return null.

What is a referenceQueue?

A ReferenceQueue is a queue that can be passed into the constructor of Reference objects. When an object wrapped by a Reference becomes garbage collected, the Reference is   enqueued onto the ReferenceQueue. This queue can then be polled for cleanup operations. If we were to subclass the SoftReference and WeakReference classes to store the name of a star, we would then have a convenient callback for removing expired map entries.

Why is a phantom reference safer than using the finalize() method?

A phantom reference in conjunction with a reference queue provides a safer alternative to the finalize() method because there is no possibility of reviving a dead object with new references.