



Glossary

A

absolute path The full path from the root directory to the file or directory, including all subdirectories that contain the file or directory.

abstract class A class cannot be instantiated and may contain abstract methods. It is marked with the `abstract` keyword.

abstract method A method that does not define an implementation when it is declared. It is marked with the `abstract` keyword. Any concrete subclass inheriting an abstract method must provide an implementation.

access modifier Defines what other code can refer to the code. Choices are `private`, `protected`, `public`, and default (no modifier).

annotation Code beginning with `@` that provides metadata about the program.

annotation element An attribute that stores values about the particular usage of an annotation.

anonymous class A nested class that does not have a name. It is declared and instantiated all in one statement using the `new` keyword.

Application Programming Interface (API) Represents code made available for your program to call.

arithmetic operator Any unary or binary value that performs a mathematical calculation. Includes `*`, `/`, `%`, `+`, `-`, `++`, and `--`.

assert statement Used to insert an assertion at a particular point in your code. It may include an optional error message.

assertion A boolean expression placed at a particular point in your code where you think something should always be true.

assignment operator An operator that assigns a value to a variable. Includes simple `=` and compound assignment operators: `+=`, `-=`, `*=`, `/=`, `%=`, `&=`, `^=`, `!=`, `<<=`, `>>=`, and `>>>=`.

asynchronous A technique where an executing thread does not wait for the results of another thread before continuing.

atomic The property of an operation to be carried out as a single unit of execution without any interference by another thread.

autoboxing When the compiler automatically converts a primitive type into the matching wrapper class.

automatic module A module that appears on the module path but does not have a `module-info.java` file.

automatic resource management When the `try-with-resources` statement calls the `close()` method on `AutoCloseable` classes without having to write custom code.

B

backed An underlying array that is changed when the `List` created from it is also changed.

binary operator An operator that takes two operands. Java includes a variety of binary operators, such as arithmetic, logical, bitwise, equality, relational, and assignment operators.

bind variable A placeholder that lets you specify the actual values at runtime in database queries.

bit Data is stored as a 0 or 1.

bitwise operator An operator that performs a bit comparison between two values. Includes `&`, `^`, and `|`.

blacklist A list of prohibited things.

block A grouping of code that is surrounded by `{}`.

bounded parameter type A generic type that specifies a bound for the generic parameter.

breadth-first search A tree-based search process that starts at the root and processes all elements of each particular depth, or distance from the root, before proceeding to the next depth.

break statement A statement within a `switch`, `while`, `do/while`, or `for` statement that causes the control flow to exit the structure and returns control to the enclosing statement.

builder pattern A creational pattern in which parameters are passed to a builder object, often through method chaining, and an object is generated with a final build call.

byte Data that is eight bits of 0s and 1s.

bytecode Compiled Java code. Bytecode appears in files with the `.class` extension.

C

case statement Defines an execution entry point for a particular value within a `switch` statement. There may be at most one `case` statement for a given value within a single `switch` statement.

casting An operator that makes a reference available via members on another class or interface.

catch block A block of code using the `catch` keyword that follows a `try` block. The catch block is often referred to as an exception handler since it can handle an exception and stop it from traversing further down the method call stack.

character encoding Determines how characters are encoded and stored in bytes in an I/O stream and later read back or decoded as characters.

checked exception An exception that must be handled or declared. It includes all exceptions that are subclasses of `java.lang.Exception` but do not inherit `java.lang.RuntimeException`.

child class When referring to two classes that share inheritance, the class that extends from another class.

circular dependency Two or more items that point to each other. This is often used with files or modules.

class A description of an object. A class in Java is defined in a source file with the `.java` extension and compiled into a file with the `.class` extension.

class variable A field within a class declared as `static`.

classpath The path on your file system where Java will search for files with the `.class` extension. The classpath can be defined by the `CLASSPATH` environment variable.

client A module that uses a service. See *consumer*.

collection The interfaces in the Java Collections Framework that represent lists, sets, and queues. Maps don't implement the `Collection` interface but are part of the Java Collection Framework.

command injection An injection attack that relies on the operating system.

comment Code that is only for documentation purposes and not executed.

compact profile In some older versions of Java, subsets of the Java libraries available for running an application.

compound assignment operator An assignment operator that first applies a numeric, logical, or shift operation to a value before assigning it to a variable. Includes `+=`, `-=`, `*=`, `/=`, `%=`, `&=`, `^=`, `!=`, `<<=`, `>>=`, and `>>>=`.

compound key A combination of columns that uniquely identify a row in a database.

concatenation Two Strings being merged together. For example, `"1" + "2"` evaluates to `"12"`.

concrete class A nonabstract class. The first concrete subclass that extends an abstract class or implements an interface is required to implement all inherited abstract methods.

concurrency The property of executing multiple threads and processes at the same time.

consistent with equals A natural ordering that uses `compareTo()`, which is said to be consistent with equals if, and only if, `x.equals(y)` is true whenever `x.compareTo(y) equals 0`.

constant Variables that do not change during the program. They use the `final` keyword.

constructor A special method within a class that gets invoked when an object is instantiated. A constructor must match the name of the class and cannot declare a return value. The purpose of a constructor is to initialize the fields of the object.

constructor overloading Multiple constructors with different signatures.

constructor reference A special type of method reference that initializes a new object.

consumer A module that uses a service; also, a functional interface that does not return a value.

containing annotation type A separate annotation that defines a `value()` array element. This is used when declaring a `@Repeatable` annotation.

context switch The process of storing a thread's current state and later restoring the state of the thread to continue execution.

continue statement A statement within a `while`, `do-while`, or `for` statement that causes the flow to finish the execution of the current loop.

control flow statement A Java statement that changes the order of execution of an application at runtime, such as those that repeat or branch to particular segments of code.

covariant return type When a method in a child class overrides a method in a parent class, the return type is permitted to be a subclass of the return type of the method defined in the parent class.

convenience methods Methods that could be implemented in other ways but that make your code easier to read and write.

copy constructor A constructor that makes a copy of the data passed into it.

cursor The position of the `ResultSet` pointer.

cycle An infinite circular dependency in which an entry in a directory is an ancestor of the directory.

cyclic dependencies Two or more items that point to each other. This is often used with files or modules. See *circular dependency*.

D

daemon thread A thread that will not prevent the JVM from exiting when the program finishes.

data Information such as a person's first or last name.

database An organized collection of information.

deadlock Occurs when two or more threads are blocked forever, each waiting on the other.

deep copy Copies an entire object structure, not just references and primitives.

default A keyword used in switch statements and interface methods. In switch statements, it represents the path taken if no case values match the input. In interfaces, it is used to declare public instance methods that include a body.

default constructor A constructor without any parameters that is generated by the compiler when no constructors are declared.

default method A method defined in an interface with the `default` keyword and includes a method body.

default package An unnamed package used when no package statement is present in the code.

defense in depth A security principle that relies on multiple techniques to protect an application.

defensive copy A security principle where the variables are copied so callers cannot change state on your object.

deferred execution Defining a block of code that will be run later.

denial of service attack A security attack when a hacker makes one or more requests with the intent of disrupting legitimate requests.

dependencies JAR files that a program depends on.

deprecation A marker to indicate that a Java element is no longer encouraged for use.

depth-first search A tree-based search process that traverses the structure from the root to an arbitrary leaf and then navigates back up toward the root, traversing fully down any paths that it skipped along the way.

deserialization The process of reconstructing an object in memory from a stored data format.

directory A record within a file system that contains files as well as other directories.

distributed denial of service attack A denial of service attack that comes from many sources at once.

do/while loop A repetition control structure that is executed at least once.

downstream collector A collector that is passed to another collector when working with streams.

E

effectively final A local variable that is not declared with the `final` keyword, but whose value is not changed after it is first set.

eligible for garbage collection An object that is no longer being accessible in a program and able to be garbage collected.

encapsulation Combining fields and methods together in a class such that the methods operate on the data, as opposed to users of the class accessing the fields directly.

enum A Java type that represents an enumeration.

enumeration A fixed set of constants.

escape The process of using one character to change the meaning of another character, often to express characters as literals that would otherwise have special meaning within a language.

exception An event that occurs during the execution of a program that disrupts the normal flow of control. In Java, an exception is an object that a method “throws” down the method call stack by handing it to the JVM.

exploit An attack that takes advantage of weak security.

extensible The ability to add functionality without recompiling the whole application.

F

factory pattern A creational pattern that uses a factory class to produce instances of objects based on a set of input parameters.

field Another name for an instance variable in a class.

FIFO First-in, first-out data structure.

file A record within a file system that stores data.

file system The system that is in charge of reading and writing data within a computer.

finally clause A block of code that follows a `try` statement and executes after the `try` and `catch` blocks, regardless of whether an exception occurs within the `try` block.

for loop A repetition control structure that includes a structure initialization, termination, and increment section. Often used to iterate over lists of items.

for-each loop An enhanced form of a `for` loop that can be applied to Java built-in arrays and objects that implement `java.lang.Iterable`, in which the loop variable structure is built and assigned automatically by the compiler.

formal type parameter Refers to the parameter used in a class, interface, method, or constructor that uses generics. The formal type parameter is declared in angle brackets. For example, `T` is the formal type parameter of the expression `<T>`.

functional interface An interface that has exactly one abstract method.

functional programming A style of programming where you declare what you want to do by writing expressions.

G

garbage collection Frees memory that is no longer used by the program.

H

handle or declare rule A rule enforced by the compiler that states if a statement throws a checked exception; the surrounding method must catch the exception or declare the exception using the `throws` keyword.

happy path The path through the code where exceptions are not thrown.

hash table A data structure with keys for lookup and values. See *map*.

heap Represents a large pool of unused memory allocated to your Java application. All objects in Java reside in the heap memory.

hidden method When a child class defines a `static` method with the same name and signature as a `static` method defined in the parent class.

hidden variable When a child class defines an instance variable with the same name as a private instance variable defined in the parent class.

high-level stream Built on top of another I/O stream using chaining.

I

identifier The name of a variable, method, class, interface, or enum.

if statement Simple form of decision control in Java in which a statement or block is executed if a boolean value evaluates to true.

immutable An object that cannot be changed.

immutable objects pattern Objects that cannot be modified after they are created.

implicit modifier A keyword such as `abstract` or `static` that is automatically applied to a member field or method regardless of whether it appears in the code.

import The Java keyword used to import a package into a source file.

inclusion attack An attack where multiple files or components are embedded within a single file.

infinite loop Code that never completes. Often occurs in a loop or recursive method.

infinite recursion A method that calls itself without ever completing.

inheritance The process by which the new child subclass automatically includes any public or protected primitives, objects, or methods defined in the parent class.

initialization block The section of a basic for loop that is executed at the start of the loop, before the first execution of the boolean expression. Variables declared within the initialization block of the for loop have limited scope and are not accessible outside the for loop.

injection attack An attack where dangerous input runs in a program.

inner class A non-static nested class defined at the member level.

instance An object created from a class.

instance variable The non-static fields in a class.

instantiation The process of bringing an Object to life. Includes calling the constructor.

integrated development environment (IDE) An editor with advanced capabilities that facilitates writing code.

interface An abstract data type that defines a list of abstract public methods that any class implementing the interface must provide. An interface can also include a list of constant variables and default methods.

internationalization The process of making an application potentially localizable. This is also known as i18n.

J

jar A Java archive file containing class files and a manifest.

JAR hell An expression representing a group of problems caused by having many and complex JAR dependencies.

Java Archive (JAR) A file that contains class files and a manifest.

Java Collections Framework A unified set of classes and interfaces defined in the `java.util` package for storing collections including `List`, `Map`, `Queue`, and `Set`.

Java Development Kit The minimum software you need in order to work with Java. It includes the `javac` and `java` commands.

Java Platform Module System The tools and techniques for organizing code into higher-level groupings than packages.

Java Virtual Machine A JVM is a virtual machine that is able to run Java bytecode class files, as well as single-file source-code programs.

JDBC Java Database Connectivity, which is the standard way to access relational databases in Java.

K

keyword Special words used by the compiler to indicate code constructs.

L

label A pointer to the head of a statement that allows the application flow to jump to it or break from it. A label is a single-word expression followed by a colon (:).

lambda expression A block of code that can be passed around. This is also known as a lambda.

lazy evaluation Code that is executed only at the time it is needed. It is often used with streams.

LIFO Last-in, first-out data structure.

list An ordered collection of elements that allows duplicate entries. Each element is accessed by an integer index.

literal A numeric, character, boolean, or `String` value that is typed into the code.

livelock Occurs when two or more threads are conceptually blocked forever, although they are each still active and trying to complete their task.

liveness The ability of an application to execute in a timely manner.

local class A nested class defined within a method, constructor, lambda expression, or initialization block.

local variable A variable declared within a method, constructor, lambda expression, or initialization block including any method parameters.

local variable type inference The formal name of the `var` feature that allows omitting the type declaration.

localization The process of supporting different locales. This is also known as `l10n`.

logical complement operator Given a boolean expression, the logical complement operator (`!`) returns the reverse of it.

logical operator An operator that compares two logical values. Includes: `&`, `^`, `|`, `&&`, and `||`.

long term support (LTS) The releases of Java that have a longer life span and are released every three years.

loop A control flow statement that repeats a particular line or block of code.

loose coupling The practice of developing classes with minimum dependencies on one another.

low-level stream An I/O stream that connects directly with the source of the data, such as a file, an array, or a string.

M

map A collection that maps keys to values with no duplicate keys allowed. The elements in a map are key/value pairs.

marker annotation An annotation without elements. Its presence is what conveys meaning.

member A field or method in a class.

member inner class See *inner class*.

memory consistency error An error that occurs when two threads have inconsistent views of what should be the same data.

metadata Data about other data, often related to the structure of a set of data.

method Also known as a function—code that can be run by name.

method declaration The definition of a Java method consisting of seven components of a method: access modifier, optional specifiers, return type, method name, parameter list, exception list, and method body.

method overloading When a class contains multiple methods with the same name but different parameter lists.

method reference As an alternative to a lambda, a method reference that states the name of a method rather than calling it.

method signature A method's name and parameter types.

module Code consisting of at least one package and a `module-info` file.

monitor A structure that supports mutual exclusion, the property that at most one thread is executing a particular segment of code at a given time. This is also called a lock.

multi-catch A feature introduced in Java 7 that allows catching multiple exception types in the same catch clause.

multiple inheritance The rule a class may inherit from multiple direct parent classes. This feature is not supported in Java.

multithreaded process A process that contains one or more threads.

mutable An object that can change state.

mutable reduction Uses the same object that can be changed to do a reduction in a pipeline.

mutual exclusion The property that at most one thread is executing a particular segment of code at a given time.

N

named module A module containing a `module-info` file and appearing on the module path.

natural ordering An attribute of classes that implement `Comparable`.

negation operator Given a numeric expression, the negation operator (`-`) returns the result of multiplying the value by `-1`.

nested class A class defined within another class. This also defines a class that is not a top-level class.

nested loop A loop that is scoped inside another loop.

numeric promotion The process by which Java will automatically upcast the data type of a smaller data type to a larger data type, as well as update an integral value to a floating-point number.

O

obfuscation The automated process of rewriting source code that purposely makes it more difficult to read.

object An instance of a class.

open source JAR files that have their source code available for users.

operand The values referenced by an operator.

operator A symbol in Java that is applied to an expression and returns a value.

operator precedence The order in which operators are evaluated.

order of operation See *operator precedence*.

overflow When a number value is too large to fit in a data type and the value automatically wraps around to the lowest possible value and counts up from there.

overriding The classification for a new method in a class that has the same signature and return type as the method in a parent class.

P

package A grouping of classes, interfaces, enumerations, and annotated types.

parallel decomposition The process of taking a task, breaking it up into smaller pieces that can be performed concurrently, and then assembling the results.

parallel reductions A reduction on a parallel stream.

parallel stream A stream that is capable of processing results concurrently, using multiple threads.

parameter The name of the variable in the method signature that is assigned the value of the argument.

parent class When referring to two classes that share inheritance, the class that is extended by another class.

partitioning A special case of grouping where there are only boolean values as grouping keys.

path A String representation of a file or directory within a file system.

path symbol A reserved series of characters that have special meaning within some file systems, such as a single dot (.) or two dots (..).

pointer Java doesn't have pointers. A reference variable referring to an Object is as close as Java gets.

polling The process of intermittently checking data at some fixed interval.

polymorphism The property of a class in Java used to take on many different forms.

post-decrement operator The -- operator when it is applied after a numeric variable. Decrements the value of the numeric variable by 1 and returns the original value.

post-increment operator The ++ operator when it is applied after a numeric variable. Increments the value of the numeric variable by 1 and returns the original value.

pre-decrement operator The -- operator when it is applied before a numeric variable. Decrements the value of the numeric variable by 1 and returns the new value.

pre-increment operator The ++ operator when it is applied before a numeric variable. Increments the value of the numeric variable by 1 and returns the new value.

primary key A unique identifier for a row in a table.

primitive types The built-in data types of the Java language. There are eight primitive types in Java: byte, short, int, long, float, double, char, and boolean.

principle of least privilege Restricts the information that code has access to.

process A group of associated threads that execute in the same shared environment.

properties file A file with the extension .properties containing key and value pairs.

Q

queue A collection that orders its elements for processing. A typical queue processes its elements in a first-in, first-out fashion, but specifying a different order is possible.

R

race condition An undesirable result that occurs when two tasks are completed at the same time, which should have been completed sequentially.

raw type A use of a generic class without specifying a generic type.

recursive method A method that calls itself to solve a problem.

reduction A special type of terminal operation where the contents of the stream are combined into a single value.

reference Variables that point to an object.

- reference types** Variables that are class types, interface types, and array types.
- reflection** An API that allows reading information about code at runtime. For example, printing a list of methods available on an object at runtime.
- regular file** A file that contains content, as opposed to a symbolic link, directory, resource, or other nonregular file that may be present in some operating systems.
- reifiable type** A type whose type information is fully known at runtime.
- relational database** A database that uses tables with rows and columns to store data.
- relational operator** An operator that compares two expressions. Includes ==, <, >, <=, >=, and instanceof.
- relative path** The path from the current working directory to a file or directory.
- reserved type name** An identifier that cannot be used for a type declaration.
- reserved word** Special words that may not be used as identifier names.
- resource** Certain types of classes such as I/O streams or JDBC objects that can be open and closed.
- resource bundle** Contains the locale-specific objects to be used by a program.
- resource leak** The result of forgetting to close resources, which can eventually cause the program to run out of memory or resources.
- return type** A primitive, reference type, or the keyword void. Defines what, if anything, is sent back from the method to the caller.
- root directory** The topmost directory in the file system, from which all files and directories inherit.
- round-robin schedule** A scheduling algorithm in which each thread receives an equal number of CPU cycles with which to execute, with threads visited in a circular order.
- runtime exception** Any exception that is a subclass of `java.lang.RuntimeException`. The handle or declare rule does not apply to runtime exceptions.

S

- scope** The portion of code where a variable can be accessed.
- search depth** The distance from the root to the current node in a search process.
- serial stream** A stream in which only one entry is processed at a time.
- serialization** The process for converting an in-memory object to a stored data format.

service In modules, the combination of an interface, any classes the interface references, and a way of looking up implementations of the interface.

service locator Responsible for finding any classes that implement a service provider interface.

service provider The implementation of a service provider interface.

service provider interface Specifies the functionality of a service.

set A collection that does not allow duplicate entries.

shallow copy Copies only the top-level primitive and reference values.

shared environment The shared space that allows threads running within the same process to share static memory and communicate directly with one another.

short-circuit operator A logical operator that may skip evaluation of the right-hand side of the expression if the value can be determined solely by the left-hand side of the equation. Includes `&&` and `||`.

signed A data type that can have a positive or negative value.

single abstract method (SAM) An interface that contains exactly one abstract method to be implemented. See *functional interface*.

single inheritance The rule that a class may inherit from only one direct parent class. Java supports single inheritance although the parent class may, in turn, inherit from another class. Java also supports implementing any number of interfaces.

single-file source-code Allows a class that does not depend on any classes outside the JDK to be run without explicitly compiling.

single-threaded process A process that contains exactly one thread.

SQL Structured Query Language, which is the language used to run queries against a relational database.

stack trace Shows the exception along with the chain of active method calls it took to get there.

starvation Occurs when a single thread is perpetually denied access to a shared resource or lock.

stateful lambda expression A lambda whose result depends on any state that might change during the execution of a pipeline.

stateless lambda expression A lambda whose result does not depend on any state that might change during the execution of a pipeline.

statement A complete unit of execution in Java, terminated with a semicolon (`;`).

static A keyword indicating the variable or method is shared by all instances of a class.

statistic A number that was calculated from data.

stored procedure Code that is compiled in advance and stored in the database.

stream Can refer to an I/O stream or a stream used in functional programming. An I/O stream is a sequence of data elements presented sequentially. A functional programming stream is used to mean a finite or infinite sequence of data.

stream operations Intermediate or terminal functions that can run on a functional programming stream.

stream pipeline A sequence of source, zero or more intermediate operations, and a terminal operation.

string pool An area in the JVM where string literals are stored. The JVM can optimize the use of string literals by allowing only one instance of a string in the pool.

subclass A class that extends a parent class.

subtype A reference to a subclass object.

superclass A class that has child classes.

supertype A reference to a superclass object.

suppressed exceptions When multiple exceptions are thrown, all but the primary exception are listed as suppressed exceptions.

swallowing an exception A scenario where you handle an exception with an empty catch block.

switch A complex decision-making structure in which a single value is evaluated and control jumps to a particular case statement or default block and continues until a break statement is reached or the switch statement finishes. Switch values may be bytes, shorts, chars, ints, enums, and Strings, as well as classes that wrap primitive data types.

symbolic link A special file within an operating system that serves as a reference to another file or directory.

synchronous The program waits for a thread to finish executing before continuing.

system thread A thread created by the JVM that runs in the background of the application, such as the garbage collector.

T

table Contains rows and columns to store data in a database.

task A single unit of work performed by a thread.

ternary operator An operator that executes one of two branches based on a conditional.

thread The smallest unit of execution that can be scheduled by the operating system.

thread pool A group of pre-instantiated reusable threads, which are available to perform a set of arbitrary tasks.

thread priority A numeric value associated with a thread that is taken into consideration by the thread scheduler when determining which threads should be currently executing.

thread safety The property of an object that guarantees safe execution by multiple threads at the same time.

thread scheduler The operating system process that determines which threads should be concurrently executing.

top-level class A class that is not an inner class.

traversing a directory The process by which you start with a parent directory and iterate over all of the descendants until some condition is met or until there are no more elements over which to iterate.

try statement A block of code containing one or more statements that may throw an exception.

try-with-resources A feature that automatically closes resources declared in the try clause.

type erasure Refers to the changes that the compiler makes to your code to remove the generics syntax and replace the generic types with `Object` references.

U

unary An operator that takes only one operand. Includes `+`, `-`, `++`, `--`, and `!`.

unboxing The compiler automatically converts a wrapper class into a primitive type.

unchecked exception An exception that is not required to be handled or declared.

underflow When a number value is too small to fit in a data type and the value automatically wraps around to the highest possible value and counts down from there.

uniform resource identifier A string of characters that identify a resource.

unnamed module A module on the classpath.

unperformed side effect The part of a short-circuit operation that might not be executed.

unsigned A data type that cannot have negative values.

update The section of a basic for loop that is executed at the end of the loop, before the boolean expression is evaluated again.

user-defined thread A thread created by the application developer to accomplish a specific task.

V

variable An allocated piece of memory for storing data. A variable has an identifier and a specific data type.

view A group of related attributes for a particular file system type.

void Used in lieu of a type when a method does not return anything.

W

while loop A repetition control structure that will continue while an expression evaluates to true or the loop is exited.

whitelist A list of allowed things.

wildcard generic type An unknown generic represented with a question mark (?).

wrapping The process by which an instance is passed to another constructor.

