1. **ENGAGE: Misconception**

**Misconception Check** In the past several years the technology evolved too fast and there’s a lot of updates in java versions that was developed and existing now. Select 2 versions of java and compare them in terms of features, advantages, and disadvantages.

Java SE 15

JDK 15 was released on September 15, 2020. Java 15 adds e.g. support for [multi-line string literals](https://en.wikipedia.org/wiki/String_literal#Multiline_string_literals) (aka Text Blocks). The Shenandoah and Z garbage collectors (latter sometimes abbreviated ZGC) are now ready for use in production (i.e. no longer marked experimental). Support for Oracle's [Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)) operating system (and SPARC CPUs) is dropped (while still available in e.g. Java 11). The Nashorn JavaScript Engine is removed. Also removed some root [CA certificates](https://en.wikipedia.org/wiki/CA_certificate).

Java SE 16

JDK 16 is the current release now under active development,[[350]](https://en.wikipedia.org/wiki/Java_version_history" \l "cite_note-350) due out in March 2021 if the current 6 month release cycles continue. The Java implementation itself was and is still written in [C++](https://en.wikipedia.org/wiki/C%2B%2B), while as of Java 16, more recent [C++14](https://en.wikipedia.org/wiki/C%2B%2B14) (but still not e.g. [C++17](https://en.wikipedia.org/wiki/C%2B%2B17)) is allowed. The code is also moved to [GitHub](https://en.wikipedia.org/wiki/GitHub" \o "GitHub) (dropping the [Mercurial](https://en.wikipedia.org/wiki/Mercurial) source control system).

1. **EXPLORE:** API Specifications list 1. Enumerate at least following API specifications of java, depending of the java version.

|  |  |
| --- | --- |
| **PACKAGES** | **DESCRIPTION** |
| [**java.awt**](https://docs.oracle.com/javase/8/docs/api/java/awt/package-summary.html) | Contains all of the classes for creating user interfaces and for painting graphics and images. |
| **Interfaces** | |
| [**ActiveEvent**](https://docs.oracle.com/javase/8/docs/api/java/awt/ActiveEvent.html) | An interface for events that know how to dispatch themselves. |
| [**Adjustable**](https://docs.oracle.com/javase/8/docs/api/java/awt/Adjustable.html) | The interface for objects which have an adjustable numeric value contained within a bounded range of values. |
| **Classes** | |
| [**AlphaComposite**](https://docs.oracle.com/javase/8/docs/api/java/awt/AlphaComposite.html) | The AlphaComposite class implements basic alpha compositing rules for combining source and destination colors to achieve blending and transparency effects with graphics and images. |
| [**AWTEvent**](https://docs.oracle.com/javase/8/docs/api/java/awt/AWTEvent.html) | The root event class for all AWT events. |
| **Exceptions** | |
| [**AWTException**](https://docs.oracle.com/javase/8/docs/api/java/awt/AWTException.html) | Signals that an Abstract Window Toolkit exception has occurred. |
| [**FontFormatException**](https://docs.oracle.com/javase/8/docs/api/java/awt/FontFormatException.html) | Thrown by method createFont in the Font class to indicate that the specified font is bad. |

|  |  |
| --- | --- |
| **PACKAGES** | **DESCRIPTION** |
| java.awt.datatransfer | Provides interfaces and classes for transferring data between and within applications. |
| **Interfaces** | |
| ClipboardOwner | Defines the interface for classes that will provide data to a clipboard. |
| FlavorListener | Defines an object which listens for FlavorEvents. |
| **Classes** | |
| Clipboard | A class that implements a mechanism to transfer data using cut/copy/paste operations. |
| DataFlavor | A DataFlavor provides meta information about data. |
| **Exceptions** | |
| MimeTypeParseException | A class to encapsulate MimeType parsing related exceptions |
| UnsupportedFlavorException | Signals that the requested data is not supported in this flavor. |

|  |  |
| --- | --- |
| **PACKAGES** | **DESCRIPTION** |
| java.awt.dnd | Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI. |
| **Interfaces** | |
| Autoscroll | During DnD operations it is possible that a user may wish to drop the subject of the operation on a region of a scrollable GUI control that is not currently visible to the user. |
| DragGestureListener | The listener interface for receiving drag gesture events. |
| **Classes** | |
| DnDConstants | This class contains constant values representing the type of action(s) to be performed by a Drag and Drop operation. |
| DragGestureEvent | A DragGestureEvent is passed to DragGestureListener's dragGestureRecognized() method when a particular DragGestureRecognizer detects that a platform dependent drag initiating gesture has occurred on the Component that it is tracking. |
| **Exceptions** | |
| InvalidDnDOperationException | This exception is thrown by various methods in the java.awt.dnd package. |

|  |  |
| --- | --- |
| **PACKAGES** | **DESCRIPTION** |
| java.awt.event | Provides interfaces and classes for dealing with different types of events fired by AWT components. |
| **Interfaces** | |
| ActionListener | The listener interface for receiving action events. |
| AdjustmentListener | The listener interface for receiving adjustment events. |
| **Classes** | |
| ActionEvent | A semantic event which indicates that a component-defined action occurred. |
| AdjustmentEvent | The adjustment event emitted by Adjustable objects like Scrollbar and ScrollPane. |

|  |  |
| --- | --- |
| **PACKAGES** | **DESCRIPTION** |
| java.awt.font | Provides classes and interface relating to fonts. |
| **Interfaces** | |
| MultipleMaster | The MultipleMaster interface represents Type 1 Multiple Master fonts. |
| MultipleMaster | The OpenType interface represents OpenType and TrueType fonts. |
| **Classes** | |
| FontRenderContext | The FontRenderContext class is a container for the information needed to correctly measure text. |
| GlyphJustificationInfo | The GlyphJustificationInfo class represents information about the justification properties of a glyph. |

**D.ELABORATE:**

Additional Details more in java API Specifications:

Give the Requirements for Writing a Java API Specifications.

This document describes the requirements for writing API specifications for the Java platform. The specification for each Java TM platform API library is made up of its Javadoc comments and additional support documentation called out in the doc comments.

This document has five sections that correspond to the sections of an API specification;

* [Top-Level Specification](https://www.oracle.com/java/technologies/javase/api-specifications.html#top-level)
* [Package Specification](https://www.oracle.com/java/technologies/javase/api-specifications.html#package)
* [Package Examples](https://www.oracle.com/java/technologies/javase/api-specifications.html#packageExample1)
* [Class/Inteface Specification](https://www.oracle.com/java/technologies/javase/api-specifications.html#class)
* [Interface Example](https://www.oracle.com/java/technologies/javase/api-specifications.html#interfaceExample1)
* [Class Examples](https://www.oracle.com/java/technologies/javase/api-specifications.html#classExample1)
* [Field Specification](https://www.oracle.com/java/technologies/javase/api-specifications.html#field)
* [Field Examples](https://www.oracle.com/java/technologies/javase/api-specifications.html#fieldExample1)
* [Method Specification](https://www.oracle.com/java/technologies/javase/api-specifications.html#method)
* [Method Examples](https://www.oracle.com/java/technologies/javase/api-specifications.html#methodExample1)