# Appendix: Shortened API documentation for exam of Application Development.

This documentation is based on docs.oracle.com/javase/8/docs/api

**java.lang**

# Class Math

## Field Summary

| | |
|---|---|
| static double | **E**<br>The double value that is closer than any other to *e*, the base of the natural logarithms. |
| static double | **PI**<br>The double value that is closer than any other to *pi*, the ratio of the circumference of a circle to its diameter. |

## Method Summary

| | |
|---|---|
| static double | **abs**(double a)<br>Returns the absolute value of a double value. |
| static double | **cos**(double a)<br>Returns the trigonometric cosine of an angle. |
| static double | **exp**(double a)<br>Returns Euler's number *e* raised to the power of a double value. |
| static double | **hypot**(double x, double y)<br>Returns sqrt($x^2$ +$y^2$) without intermediate overflow or underflow. |
| static double | **log**(double a)<br>Returns the natural logarithm (base *e*) of a double value. |
| static double | **pow**(double a, double b)<br>Returns the value of the first argument raised to the power of the second argument. |
| static double | **random**()<br>Returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0. |
| static int | **round**(float a)<br>Returns the closest int to the argument. |
| static double | **sin**(double a)<br>Returns the trigonometric sine of an angle. |
| static double | **sqrt**(double a)<br>Returns the correctly rounded positive square root of a double value. |
| static double | **tan**(double a)<br>Returns the trigonometric tangent of an angle. |

**java.awt**

# Class Graphics

## Constructor Summary

| | |
|---|---|
| protected | **Graphics**()<br>Constructs a new Graphics object. |

## Method Summary

| | |
|---|---|
| abstract void | **drawArc**(int x, int y, int width, int height, int startAngle, int arcAngle)<br>Draws the outline of a circular or elliptical arc covering the specified rectangle. |
| abstract void | **drawLine**(int x1, int y1, int x2, int y2)<br>Draws a line, using the current color, between the points (x1, y1) and (x2, y2) in this graphics context's coordinate system. |
| abstract void | **drawOval**(int x, int y, int width, int height)<br>Draws the outline of an oval. |
| void | **drawRect**(int x, int y, int width, int height)<br>Draws the outline of the specified rectangle. |
| abstract void | **drawString**(AttributedCharacterIterator iterator, int x, int y)<br>Draws the text given by the specified iterator, using this graphics context's current color. |
| abstract void | **drawString**(String str, int x, int y)<br>Draws the text given by the specified string, using this graphics context's current font and color. |
| abstract void | **fillArc**(int x, int y, int width, int height, int startAngle, int arcAngle)<br>Fills a circular or elliptical arc covering the specified rectangle. |
| abstract void | **fillOval**(int x, int y, int width, int height)<br>Fills an oval bounded by the specified rectangle with the current color. |
| abstract void | **fillRect**(int x, int y, int width, int height)<br>Fills the specified rectangle. |
| abstract Color | **getColor**()<br>Gets this graphics context's current color. |
| abstract void | **setColor**(Color c)<br>Sets this graphics context's current color to the specified color. |

**java.util**

# Class ArrayList

## Constructor Summary

| | |
|---|---|
| **ArrayList**() | Constructs an empty list with an initial capacity of ten. |
| **ArrayList**(int initialCapacity) | Constructs an empty list with the specified initial capacity. |

## Method Summary

| | |
|---|---|
| boolean | **add**(E e)<br>Appends the specified element to the end of this list. |
| void | **add**(int index, E element)<br>Inserts the specified element at the specified position in this list. |
| void | **clear**()<br>Removes all of the elements from this list. |
| boolean | **contains**(Object o)<br>Returns `true` if this list contains the specified element. |
| E | **get**(int index)<br>Returns the element at the specified position in this list. |
| int | **indexOf**(Object o)<br>Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element. |
| boolean | **isEmpty**()<br>Returns `true` if this list contains no elements. |
| E | **remove**(int index)<br>Removes the element at the specified position in this list. |
| boolean | **remove**(Object o)<br>Removes the first occurrence of the specified element from this list, if it is present. |
| E | **set**(int index, E element)<br>Replaces the element at the specified position in this list with the specified element. |
| int | **size**()<br>Returns the number of elements in this list. |

**java.lang**

# Class String

## Method Summary

| | |
|---|---|
| char | **charAt**(int index)<br>Returns the char value at the specified index. |
| int | **compareTo**(String anotherString)<br>Compares two strings lexicographically. |
| int | **compareToIgnoreCase**(String str)<br>Compares two strings lexicographically, ignoring case differences. |
| String | **concat**(String str)<br>Concatenates the specified string to the end of this string. |
| boolean | **contains**(CharSequence s)<br>Returns true if and only if this string contains the specified sequence of char values. |
| boolean | **endsWith**(String suffix)<br>Tests if this string ends with the specified suffix. |
| boolean | **equals**(Object anObject)<br>Compares this string to the specified object. |
| boolean | **equalsIgnoreCase**(String anotherString)<br>Compares this String to another String, ignoring case considerations. |
| int | **indexOf**(int ch)<br>Returns the index within this string of the first occurrence of the specified character. |
| int | **indexOf**(String str)<br>Returns the index within this string of the first occurrence of the specified substring. |
| boolean | **isEmpty**()<br>Returns true if, and only if, length() is 0. |
| int | **lastIndexOf**(int ch)<br>Returns the index within this string of the last occurrence of the specified character. |
| int | **lastIndexOf**(String str)<br>Returns the index within this string of the rightmost occurrence of the specified substring. |
| int | **length**()<br>Returns the length of this string. |
| boolean | **startsWith**(String prefix)<br>Tests if this string starts with the specified prefix. |
| String | **substring**(int beginIndex)<br>Returns a new string that is a substring of this string. |
| String | **substring**(int beginIndex, int endIndex)<br>Returns a new string that is a substring of this string. |
| String | **toUppercase**()<br>Converts all of the characters in this String to upper case using the rules of the default locale. |

**javax.swing**

# Class JTextField

```
public class JTextField
extends JTextComponent
```

## Method Summary

| | |
|---|---|
| void | **copy**()<br>Transfers the currently selected range in the associated text model to the system clipboard, leaving the contents in the text model. |
| void | **cut**()<br>Transfers the currently selected range in the associated text model to the system clipboard, removing the contents from the model. |
| int | **getSelectionEnd**()<br>Returns the selected text's end position. |
| int | **getSelectionStart**()<br>Returns the selected text's start position. |
| String | **getText**()<br>Returns the text contained in this TextComponent. |
| String | **getText**(int offs, int len)<br>Fetches a portion of the text represented by the component. |
| boolean | **isEditable**()<br>Returns the boolean indicating whether this TextComponent is editable or not. |
| void | **setEditable**(boolean b)<br>Sets the specified boolean to indicate whether or not this TextComponent should be editable. |
| void | **setSelectedTextColor**(Color c)<br>Sets the current color used to render the selected text. |
| void | **setSelectionColor**(Color c)<br>Sets the current color used to render the selection. |
| void | **setSelectionEnd**(int selectionEnd)<br>Sets the selection end to the specified position. |
| void | **setSelectionStart**(int selectionStart)<br>Sets the selection start to the specified position. |
| void | **setText**(String t)<br>Sets the text of this TextComponent to the specified text. |

**javax.swing**

# Class JButton

```
public class JButton
extends AbstractButton
```

## Method Summary

| | |
|---|---|
| String | **getText**()<br>Returns the button's text. |
| void | **setText**(String t)<br>Sets the button's text. |
| boolean | **isSelected**()<br>Returns the state of the button. True if the toggle button is selected, false if it's not. |
| void | **setIcon**(Icon defaultIcon)<br>Sets the button's default icon. This icon is also used as the "pressed" and "disabled" icon if there is no explicitly set pressed icon. |

**java.lang**

# Class Integer

## Method Summary

| | |
|---|---|
| int | **compareTo**(Integer anotherInteger)<br>Compares two Integer objects numerically. |
| double | **doubleValue**()<br>Returns the value of this Integer as a double. |
| boolean | **equals**(Object obj)<br>Compares this object to the specified object. |
| int | **intValue**()<br>Returns the value of this Integer as an int. |
| static int | **parseInt**(String s)<br>Parses the string argument as a signed decimal integer. |
| static int | **parseInt**(String s, int radix)<br>Parses the string argument as a signed integer in the radix specified by the second argument. |
| String | **toString**()<br>Returns a String object representing this Integer's value. |
| static String | **toString**(int i)<br>Returns a String object representing the specified integer. |
| static String | **toString**(int i, int radix)<br>Returns a string representation of the first argument in the radix specified by the second argument. |

**java.lang**

# Class Double

## Method Summary

| | |
|---|---|
| int | **compareTo**(Double anotherDouble)<br>Compares two Double objects numerically. |
| double | **doubleValue**()<br>Returns the double value of this Double object. |
| boolean | **equals**(Object obj)<br>Compares this object to the specified object. |
| int | **intValue**()<br>Returns the value of this Double as an int (by casting to type int). |
| static double | **parseDouble**(String s)<br>Returns a new double initialized to the value represented by the specified String, as performed by the valueOf method of class Double. |
| String | **toString**()<br>Returns a string representation of this Double object. |
| static String | **toString**(double d)<br>Returns a string representation of the double argument. |