SUMMARY OF JAVA STANDARD LIBRARY METHODS FOR SELECTED CLASSES

- `String` (package `java.lang`) Methods:
  - `public String(char[] value)`: Allocates a new `String` so that it represents the sequence of characters currently contained in the character array `value`.
  - `public int length()`: Returns the length of this `String`.
  - `public char charAt(int index)`: Returns the `char` value at the specified `index`.
  - `public char[] toCharArray()`: Converts this `String` to a new character array.
  - `public boolean equals(Object anObject)`: Compares this `String` to `anObject`.
  - `public boolean equalsIgnoreCase(String anotherString)`: Compares, ignoring case considerations, this `String` to `anotherString`.
  - `public int compareTo(String anotherString)`: Compares this `String` to `anotherString` lexicographically; returns a negative value if this `String` occurs before `anotherString`, a positive value if this `String` occurs after `anotherString`, and `0` if both `String`s are equal.
  - `public int compareToIgnoreCase(String anotherString)`: Compares, ignoring case considerations, this `String` to `anotherString` lexicographically; returns a negative value if this `String` occurs before `anotherString`, a positive value if this `String` occurs after `anotherString`, and `0` if both `String`s are equal.
  - `public int indexOf(int ch)`: Returns the index within this `String` of the first occurrence of character `ch`, `-1` if it does not occur.
  - `public int indexOf(int ch, int fromIndex)`: Returns the index within this `String` of the first occurrence of character `ch`, starting the search at position `fromIndex`; returns `-1` if `ch` does not occur in this `String`.
  - `public int indexOf(String str)`: Returns the index within this `String` of the first occurrence of substring `str`, `-1` if it does not occur.
  - `public int indexOf(String str, int fromIndex)`: Returns the index within this `String` of the first occurrence of substring `str`, starting at position `fromIndex`; returns `-1` if `str` does not occur in this `String`.
  - `public String substring(int beginIndex)`: Returns a new `String` which is a substring of this `String`, composed of the characters starting at position `beginIndex` (inclusive).
  - `public String substring(int beginIndex, int endIndex)`: Returns a new `String` that is a substring of this `String`, composed of the characters starting at position `beginIndex` (inclusive), and ending at position `endIndex` (exclusive).
  - `public String replace(char oldChar, char newChar)`: Returns a new `String` resulting from replacing all occurrences of `oldChar` in this `String` with `newChar`.
  - `public String toLowerCase()`: Returns a new `String` consisting of all the characters in this `String` converted to lower case.
  - `public String toUpperCase()`: Returns a new `String` consisting of all the characters in this `String` converted to upper case.
  - `public String trim()`: Returns a copy of this `String`, with leading and trailing whitespace omitted.
- `Scanner` (package `java.util`) Methods:
  - `public Scanner(File source) throws java.io.FileNotFoundException`: Constructs a new `Scanner` that produces values scanned from the specified file.
  - `public Scanner(InputStream source)`: Constructs a new `Scanner` that produces values scanned from the specified input stream.
  - `public Scanner(String source)`: Constructs a new `Scanner` that produces values scanned from the specified `String`.
  - `public void close()`: Closes this `Scanner`.
  - `public boolean hasNext()`: Returns `true` if this `Scanner` has another token in its input.
  - `public boolean hasNextDouble()`: Returns `true` if the next token in this `Scanner`'s input can be interpreted as a `double` value using the `nextDouble()` method.
  - `public boolean hasNextInt()`: Returns `true` if the next token in this `Scanner`'s input can be interpreted as an `int` value using the `nextInt()` method.
  - `public boolean hasNextLine()`: Returns `true` if there is another line in the input of this `Scanner`

- `public boolean hasNextLong()`: Returns `true` if the next token in this `Scanner`'s input can be interpreted as a `long` value using the `nextLong()` method.
- `public String next()`: Finds and returns the next complete token from this `Scanner`.
- `public double nextDouble()`: Scans the next token of the input as a `double`.
- `public int nextInt()`: Scans the next token of the input as an `int`.
- `public String nextLine()`: Advances this `Scanner` past the current line and returns the input read.
- `public int nextLong()`: Scans the next token of the input as an `long`.

- `PrintStream` (package `java.io`) Methods:

  - `public PrintStream(File file) throws java.io.FileNotFoundException`: Creates a new `PrintStream` which writes to the specified `File`.
  - `public PrintStream(String fileName) throws java.io.FileNotFoundException`: Initializes a new `PrintStream` which writes to the file with the specified `fileName`.
  - `public void close()`: Closes the stream.
  - `public void print(boolean b)`: Prints `boolean` value `b`.
  - `public void print(char c)`: Prints `char` value `c`.
  - `public void print(char[] s)`: Prints the array of `char` `s`.
  - `public void print(double d)`: Prints `double` value `d`.
  - `public void print(int i)`: Prints `int` value `i`.
  - `public void print(Object o)`: Prints `Object` `o`.
  - `public void print(String s)`: Prints `String` `s`.
  - `public void println()`: Terminates the current line by writing the line separator string.
  - `public void println(boolean b)`: Prints `boolean` value `b` and then terminates the line.
  - `public void println(char c)`: Prints `char` value `c` and then terminates the line.
  - `public void println(char[] s)`: Prints array of `char` `s` and then terminates the line.
  - `public void println(double d)`: Prints `double` value `d` and then terminates the line.
  - `public void println(int i)`: Prints `int` value `i` and then terminates the line.
  - `public void println(Object o)`: Prints `Object` `o` and then terminates the line.
  - `public void println(String s)`: Prints `String` `s` and then terminates the line.

  Note that the `PrintWriter` class defines the same methods and constructors (except for the fact that the constructors are called `PrintWriter` instead of `PrintStream`).

- `Math` (package `java.lang`) Methods:

  - `public static double pow(double a, double b)`: Returns the value of `a` raised to the power of `b`.
  - `public static double sqrt(double a)`: Returns the correctly rounded positive square root of `double` value `a`.
  - `public static double random()`: Returns a `double` value with a positive sign, greater than or equal to `0.0` and less than `1.0`.
  - `public static double sin(double a)`: Returns the trigonometric sine of angle `a`, where `a` is in radians.
  - `public static double cos(double a)`: Returns the trigonometric cosine of angle `a`, where `a` is in radians.
  - `public static double tan(double a)`: Returns the trigonometric tangent of angle `a`, where `a` is in radians.
  - `public static double toDegrees(double angrad)`: Converts angle `angrad` measured in radians to an approximately equivalent angle measured in degrees.
  - `public static double toRadians(double angdeg)`: Converts angle `angdeg` measured in degrees to an approximately equivalent angle measured in radians.
  - `public static double exp(double a)`: Returns Euler's number $e$ raised to the power of `double` value `a`.
  - `public static double log(double a)`: Returns the natural logarithm (base $e$) of `double` value `a`.
  - `public static double log10(double a)`: Returns the base 10 logarithm of `double` value `a`.

- `Character` (package `java.lang`) Methods:

  - `public static boolean isDigit(char ch)`: Determines if character `ch` is a digit.
  - `public static int digit(char ch, int radix)`: Returns the numeric value of character `ch` in the radix `radix`, `-1` if `ch` does not represent a digit.
  - `public static char forDigit(int digit, int radix)`: Returns the character representation of `digit` in the radix `radix`.
  - `public static boolean isLetter(char ch)`: Determines if character `ch` is a letter.
  - `public static boolean isLowerCase(char ch)`: Determines if character `ch` is a lowercase character.
  - `public static boolean isUpperCase(char ch)`: Determines if character `ch` is an uppercase character.
  - `public static boolean isWhitespace(char ch)`: Determines if character `ch` is white space according to Java.
  - `public static char toLowerCase(char ch)`: Converts character `ch` to lowercase.
  - `public static char toUpperCase(char ch)`: Converts character `ch` to uppercase.

- `ArrayList<E>` (package `java.util`) Methods:

  - `public ArrayList<E>()`: Creates a new empty `ArrayList` which contains elements of type `E`.
  - `public int size()`: Returns the number of elements in this list.
  - `public boolean isEmpty()`: Returns `true` if this list contains no elements.
  - `public boolean contains(Object o)`: Returns `true` if this list contains element `o`; comparisons are performed using the `equals()` method on `o`.
  - `public int indexOf(Object o)`: Returns the index of the first occurrence of element `o` in this list, or `-1` if this list does not contain this element; comparisons are performed using the `equals()` method on `o`.
  - `public E get(int index)`: Returns the element at position `index` in this list.
  - `public E set(int index, E element)`: Replaces the element at the position `index` in this list with the specified `element`.
  - `public boolean add(E e)`: Appends the specified `element` to the end of this list.
  - `public void add(int index, E element)`: Inserts the specified element at the position `index` in this list.
  - `public E remove(int index)`: Removes the element at position `index` in this list.
  - `public boolean remove(Object o)`: Removes the first occurrence of the specified element `o` from this list, if it is present; comparisons are performed using the `equals()` method on `o`.
  - `public void clear()`: Removes all of the elements from this list.

- `File` (package `java.io`) Methods:

  - `public File(String pathname)`: Creates a `File` representing the file at the given `pathname`.

DESCRIPTIONS OF CLASSES AND METHODS PROVIDED SPECIFICALLY FOR THE EXAMINATION, IF ANY, WILL BE LISTED HERE