Java API Documentation for Exam 2 (with ArrayList) ⤴

Excerpts from the Java API Documentation

Excerpt from documentation for java.util.ArrayList<E>

boolean	add(E element) Appends the specified element to the end of this list.
boolean	add(int index, E element) Appends the specified element at the specified position in this list.
void	clear() Removes all of the elements from this list.
boolean	contains(E element) Returns true if this list contains the specified element.
E	get(int index) Returns the element at the specified position in this list.
E	remove(int index) Removes the element at the specified position in this list (elements to right shift down)
boolean	remove(Object obj) Removes the first occurrence of the specified object, returning false if it does not occur
E	set(int index, E element) Replaces the element at the given position in this list with the specified element.
int	size() Returns the number of elements in this list.

Excerpt from documentation for java.lang.String

char	charAt(int index) Returns the char at the specified index.
boolean	equals(String anotherString) Returns true if this string is the same as the given string.
int	indexOf(char ch) Returns the index within this string of the first occurrence of the given character, or -1 if the character does not occur in this string.
int	indexOf(String str) Returns the index within this string of the first occurrence of the given substring, or -1 if the given substring does not occur in this string.
int	lastIndexOf(char ch) Returns the index within this string of the last occurrence of the given character, or -1 if the character does not occur in this string.
int	length() Returns the length of this string
String	substring(int beginIndex) Returns a substring of this string starting at beginIndex
String	substring(int beginIndex, int endIndex) Returns a substring consisting of characters from beginIndex through endIndex - 1
String[]	split(String regex) Splits this string using the given string expression as the delimiter.
String	trim() Returns a copy of this string with leading and trailing whitespace removed

Excerpt from documentation for java.lang.Integer

int	parseInt(String s) Parses the string argument as an integer.

Excerpt from documentation for java.lang.Double

double	parseDouble(String s) Parses the string argument as an floating-point number.

Excerpt from documentation for java.util.Random

int	`nextInt(int n)`
	Returns a pseudorandom, uniformly distributed int value between 0 (inclusive) and the specified value (exclusive)

java.util.Random constructor

`Random()`
Creates a new random number generator.

Excerpt from documentation for java.io.File

String	`getName()`
	Returns the name of the file or directory represented by this object.
boolean	`isDirectory()`
	Returns true if this object represents a directory.
File[]	`listFiles()`
	If this object represents a directory, returns a list of File objects representing its immediate contents; otherwise returns null

java.io.File constructor

`File(String filename)`
Constructs a new File instance.

Excerpt from documentation for java.util.Scanner

void	`close()`
	Closes this scanner and associated input stream, if any.
boolean	`hasNext()`
	Returns true if this scanner has another token in its input.
boolean	`hasNextDouble()`
	Returns true if the next token in this scanner's input can be interpreted as a double value
boolean	`hasNextInt()`
	Returns true if the next token in this scanner's input can be interpreted as an int value.
boolean	`hasNextLine()`
	Returns true if there is another line in the input of this scanner.
String	`next()`
	Finds and returns the next complete token from this scanner.
double	`nextDouble()`
	Scans the next token of the input as a double.
int	`nextInt()`
	Scans the next token of the input as an int.
String	`nextLine()`
	Returns all input up to the next line break and advances to the next line.

java.util.Scanner constructors

`Scanner(File source)`
Constructs a new Scanner that provides values scanned from the specified file.
`Scanner(InputStream source)`
Constructs a new Scanner that provides values scanned from the specified input stream, such as System.in.
`Scanner(String source)`
Constructs a new Scanner that provides values scanned from the specified String.