Ada vs. Java

Ada	Java
General	
to compile and run hello.adb gnatmake hello hello	to compile and run hello.java javac hello.java java hello
-- comment	// comment /* comment */
Ada is NOT case sensitive	Java is case sensitive
File name must match name of package or procedure in file (file name must be lower case in gnat).	File name must match name of single public class in file
with *package*	all classes in classpath searched automatically
use *MyPackage*	import MyPackage.*
no garbage collection (in gnat)	garbage collection
Statements	
i := 1;	i = 1;
i := i + 1;	i = i + 1; i += 1; i++; ++i;

if *condition* then
  statements
elsif
  statements
else
  statements
end if;
	if *(condition){*
	  statements
	}
	else if *(condition) {*
	  statements
	}
	else{
	  statements
	}

for i in 1 .. 10 loop
  statements
end loop
	for (i=1; i<=10; i++) {
	  statements
	}

while i < 10 loop
  statements
end loop
	while (i<10){
	  statements
	}

exit; -- leave a loop	break; // leave a loop
exit when *condition*;	if *(condition)* break;

case i of
  when 3 => *statement*
  when 4 => *statement*
  when others => *statement*
end case;
	switch (i){
	  3: *statement*
	    break
	  4: *statement*
	    break
	  else: *statement* // not required
	}

i := getit; -- parameter less function	i = getit(); // All methods need parentheses
null;	;
OPERATORS	
arithmetic: +, -, *, / relational: <, <=, >, >=	same
relational: =, /=	==, !=
boolean: and, or, xor, not	boolean: &, |, ^, !
div, mod	%, /
operators can be overloaded	operators can not be overloaded

Ada	Java
TYPES	
Integer -- Size dependent on machine	byte, short, int, long (8, 16, 32 and 64 bits respectively)
Natural	unsigned
float	float (32 bits), double (64 bits)
character -- 8 bit ascii wide_character -- 16 bit unicode	char // 16 bit unicode
String -- predefined type	String // built in class
boolean: true, false	boolean: true, false
subrange: 1 .. 10	none
enumerated: (red, blue)	None: use integers
type ... is ...	Classes are the only new types
DECLARATIONS	
All declarations before begin	Declarations allowed anywhere
i, j: Integer;	int i, j;
i: Integer := 3; j, k: Integer := 4; -- j is 4	int i=3; int j, k = 4; // does not initialise j
c: constant integer := 5;	final int c = 5;
a: array [0..9] of Integer;	int[] a = new int[10]; // all arrays begin at 0
a: array [0..2] of Integer := (5,6,7);	int[] a = {5,6,7};

Array attributes:
'Length
'First
'Last
'Range
	Array members:
	length()
	// all arrays begin at 0
	.length()-1
	// no range

type MyRec is record
   i: integer;
   c: character;
end record;
R: MyRec;
	class MyClass{
	   int i;
	   char c;
	}
	MyClass X = new MyRec();

Only get pointers when specified	All class types are reference types
FUNCTIONS, PROCEDURES	**METHODS**

function hello(i, j: integer) return integer is
begin
   statements
end hello;
	int hello(int i, int j){
	  statements
	}

procedure hello(i: integer) is
begin
   statements
end hello;
	void hello(int i){
	  statements
	}

procedure hello(i: out integer) is	No out parameters in java
EXCEPTIONS	

begin
    statements1
exception
  when *an_exception* => *statements2*
   when others => *statements3*
end
	try{
	  statements1
	} catch (*AnException p)*{
	  statements2
	} catch (Exception e){
	  statements3
	}
	// Alternative
	int hello() throws *SomeException{*
	  statements // without try/catch
	}