UA

Sistemas de Operação / Fundamentos de Sistemas Operativos

Brief comparison of Java and C/C++

Artur Pereira <artur@ua.pt>

DETI / Universidade de Aveiro

Outline

Introduction
Objectives

- I don't want to teach programming
- I just want to point out some tips that can help to bridge the gap between Java and C/C++

How?

- Pointing out correspondences between elements of the two languages
- Pointing out elements that exist in one and not in the other

Variables
Data types (1)

Integer numbers

Java	C/C++
`byte`, `short`, `int`, `long`	`char`, `short`, `int`, `long`
only signed	`signed`, `unsigned`
fixed size	non-fixed size
	`#include <stdint.h>`

Real numbers

Java	C/C++
`float`, `double`	`float`, `double`

Booleans

Java	C/C++
`boolean`	`bool` (C++)
	`#include <stdbool.h>` (C)

Variables
Data types (2)

Characters

Java	C/C++
• `char` (unicode)	• `char` (8-bit interger)

References

Java	C/C++
• variables of non primitive types are references to objects	• pointers (`*`, `&`) • references (`&`) (C++)

Tuples

Java	C/C++
• `class` • Can contain methods • nesting by reference	• `class` (C++) • Can contain methods (C++) • nesting by value / reference • `struct` (public)

Variables
Data types (3)

Arrays

Java	C/C++
• The `[]` operator • An Object • The `length` property	• The `[]` operator • A pointer to an address location • The `sizeof` operator (in some cases)

Strings

Java	C/C++
• `String` class • An Object, sequence of `char` (Unicode code points) • immutable • Operator `+`	• An array • A null-terminated sequence of `char` (8-bit interger)

Variables
Lifetime

static variables

Java	C/C++
• class attributes (`static`)	• class/struct attributes (`static`) (only C++) • locally global attributes (`static`) • global attributes

dynamic (heap) variables

Java	C/C++
• `new` operator • implicit deletion • only Object and derivatives	• `new` operator • explicit deletion (`delete` operator) • any type

automatic (stack) variables

Java	C/C++
• only primitive types (including references)	• any type

Variables
Operators (1)

Arithmetic operators

Java	C/C++
• `+, -, *, /, %`	• `+, -, *, /, %`

Relational operators

Java	C/C++
• `>, <, >=, <=, ==, !=`	• `>, <, >=, <=, ==, !=`

Logical operators

Java	C/C++
• `&&, ||, !`	• `&&, ||, !` • `and, or, not` (C++)

Variables
Operators (2)

Bitwise operators

Java	C/C++
`&`, `|`, `~` , `^`	`&`, `|`, `~` , `^`
`<<`, `>>`, `>>>`	`<<`, `>>`

Assignment operators

Java	C/C++
`=`	`=`
`+=`, `-=`, `*=`, `/=`, `%=`	`+=`, `-=`, `*=`, `/=`, `%=`
`&=`, `|=`, `^=`	`&=`, `|=`, `^=`
`<<=`, `>>=`, `>>>=`	`<<=`, `>>=`

Variables
Operators (3)

Ternary operator

Java	C/C++
`?` `:`	`?` `:`

accessing tuple fields

Java	C/C++
`.`	`.`, `->`
	`&`, `*`
	`::` (C++)

Statements
Flux control

Conditional statements

Java	C/C++
• if-then-else	• if-then-else
• switch-case	• switch-case

Loop statements

Java	C/C++
• for	• for
• while	• while
• do-while	• do-while
• for-each	• for-each (C++)

Break statements

Java	C/C++
• `break`, `continue`	• `break`, `continue`
• `break` label	

Object-oriented features

inheritance

Java	only C++
• simple inheritance (`Object`)	• multiple inheritance
• `extends` operator	• `:` operator

access modifiers

Java	only C++
• `public`	• `public` (default in `struct`)
• `protected`	• `protected`
• `private`	• `private` (default in `class`)
• package (default)	

Object-oriented features

polymorphism

Java	only C++
• function overloading	• function overloading
• function overriding	• function overriding
• `final`	• `virtual`
	• operator overloading

Input/Output

Libraries

Java	C/C++
• `import Java.io`	• `#include <stdio.h>`
	• `#include <iostream>` (C++)

Standard streams

Java	C/C++
• `System.in`	• `stdin`
	• `std::cin` (C++)
• `System.out`	• `stdout`
	• `std::cout` (C++)
• `System.err`	• `stderr`
	• `std::cerr` (C++)

Functions

Type of functions

Java	C/C++
• methods in classes	• methods in classes (C++) • global functions • locally global functions (`static`) • functions within namespaces (C++)

Arguments to functions

Java	C/C++
• by value (primitive type) • by reference (Object type)	• by value (any type) • by pointer (any type) • by reference (any type) (C++) • default values (C++)