The 5W1H of D Programming Language

2005/10/06

稲葉 一浩 (Kazuhiro Inaba)
http://www.kmonos.net/
What is D?
What is D?

- Multi-Paradigm Language
  - Object-Oriented Programming
  - Generic Programming (via templates)
- Compiled
  - Native code
  - No VM, No Interpreters
- Statically-Typed
What is D?

- Garbage Collected
- Unicode Based
- Binary Compatible with C
- Looks and Feels very much like C/C++
Code Example: Hello World

```c
import std.cstream;

int main( char[][[]] args )
{
    dout.writeln( "Hello, World!" );
    return 0;
}
```
interface Animal {
    char[] speak();
}

class Dog : Animal {
    char[] speak() { return "woof"; }
}

class Cat : Animal {
    char[] speak() { return "meow"; }
}
int main()
{
    Animal[] animals;
    animals ~= new Dog();
    animals ~= new Cat();

    foreach( Animal a ; animals )
        dout.writeLine( a.speak() );
    return 0;
}
Class

- Encapsulation
  - public, protected, private

- Inheritance
  - Single Inheritance
    - Multiple Interfaces
    - Mix-in
  - ‘class Object’ at the root of the hierarchy

Pretty Much Like Java & C#! (except mix-ins)
Class

- Property (like C#)
  - Setter/Getter as if it were a field

- Operator Overloading (like C++, C#)
  - \( a + b \rightarrow a\text{.opAdd}(b) \)
class Stack(T) {
    private T[] data;
    public void push(T e) {
        data ~= e;
    }

    public T pop() {
        T e = data[$-1];
        data.length = data.length-1;
        return e;
    }
}

Code Example: Templates

```c
int main()
{
    Stack!(int) s = new Stack!(int)();

    s.push(100);
    s.push(200);
    dout.writefln( s.pop() ); // 200
    dout.writefln( s.pop() ); // 100
    return 0;
}
```
Templates

- Set of Parameterized Declarations
  - class
  - function
  - variable
  - typedef ...

Similar to C++ Templates (more powerful)
What is D?

- Reengineering of C/C++
  - Native-code compiler
  - Binary Compatible with C
  - Familiar Look&Feel
- Incorporating many features of modern languages: Java, C#, ...
  - GC
  - Modules
  - OO Based on Single Inheritance
- ... And more!
When was D born?
When was D born?

- 1999 Dec
  - Conceived by Walter Bright
- 2001 Aug
  - Language spec draft published
- 2001 Dec
  - First Alpha Version of D Compiler (DMD)
- 2004 Mar
  - GDC – D Front End for GCC
Constantly developed

Still in Beta, Still Slightly Evolving
Who created D?
Who created D?

- Walter Bright
  - The author of ...
    - Northwest Software C
    - Datalight C
    - Zorland C
    - Zortech C++ (the first native C++ compiler)
    - Symantec C++
    - Symantec Visual Cafe for Java
    - Digital Mars C++
  - Genuine Compiler Hacker!
Who created D?

- Walter Bright – Compiler Hacker
  - Emphasizes the “Easiness of Compiler Implementation” of D Language, which leads to...
    - Ultra-fast compilation time
    - Easiness for implementation of other tools
      - Syntax Highlighting
      - Code Completion
      - Refactoring tools
Why D?
Why D?

- What’s different from C/C++/Java/C#?
  - Powerful Built-in Arrays and Strings
  - Built-in Complex Numbers
  - ‘with’ statement
  - Nested Functions and Delegates
  - Mix-in
  - RAII, Contract Programming, Unittest
Array Slice, Concat, ...

```java
char[] s = "Hello";
char[] t = s[1..3];  // "el"
char[] u = s ~ t;    // "Helloel"
u[4..$] = '_';       // "Hell___"
```
int main( char[][] args )
{
    foreach( char[] arg ; args )
        switch( arg )
        {
            case "-h"
            case "--help"
            case "-i"
            default:
                break;
        }
}
Built-in Associative Arrays

```c
long[char[]] pfx;

pfx[“Kilo”] = 1_000;
pfx[“Mega”] = 1_000_000;
pfx[“Giga”] = 1_000_000_000;
pfx[“Tera”] = 1_000_000_000_000;

if( !(“Peta” in pfx) )
    pfx[“Peta”] = pfx[“Tera”]*1000;
```
‘with’ statement

```java
with(dout)
{
    writeln(“Hello”);
    writeln(“World”);
}
```
Nested Function

```c
void f()
{
    int x=0;
    void g() { ++x; }
    g(); g(); g();
}
```
Delegates

OutputStream delegate(...) p = 
&dout.writeln;

p(“Hello”);
p(100);
p(-5.3+4i);
void loop(int n, void delegate() s) {
    for(int i=0; i<n; ++i)
        s();
}

loop( 10,
    delegate{dout.writeln("Hi!");} );
Anonymous Delegates

```cpp
void loop(int n,
          void delegate(int i) s)
{
    for(int i=0; i<n; ++i)
        s(i);
}

int x=0;
loop(10, delegate(int i){x+=i;});
```
Mix-in

template Debug() {
    void whoAmI() {
        TypeInfo t = typeid(typeof(this));
        dout.writeln( t, ― – ‖, this );
    }
}

class Dog {
    mixin Debug!();
    char[] toString() {
        Dog d = new Dog("Pochi");
        d.whoAmI(); //Dog – Pochi
double sqrt( double x )
in
  { assert( x>=0 ); }
out(r)
  { assert( abs(r*r-x)<0.0001 ); }
body
  {
    ...
  }
Builtin Unittests

char toUpper( char c ) {
    return ‘a’<=c && c<='z'
        ? c-'a'+'A' : c;
}

unittest {
    assert( toUpper(‘a’) == ‘A’ )
    assert( toUpper(‘T’) == ‘T’ )
}
Where is D used?
Where is D used?

- Supported Systems
  - DMD – Digital Mars D Compiler
    - Windows, Linux on x86
  - GDC – GNU D Compiler
    - Linux, FreeBSD, MacOSX, Cygwin, AIX, ...
Where is D used?

- Applications written in D
  - Games!
  - Demos!
  - SDL (Simple DirectMedia Layer)
ABA
isshiki
shinichiro.h
yaneurao
nekokaneko               Blacker
Where is D used?

- Other Applications written in D
  - akIDE
    - An IDE targetting D and written in D
  - attoHttpd
    - Simple Http Server
  - delmail
    - Spam-mail killer
  - Diki
    - Simple wiki engine
  - DMDScript
    - ECMAScript interpreter
How to get D?
How to get D?

- Just Google it! 😊
How to get D?

- Or, use package systems
  - FreeBSD: /usr/ports/lang/gdc
  - Cygwin: Devel >> gcc-gdc
Thank you for listening!