

D Programming Language

Czy można połączyć wydajność z produktywnością?



Typy podstawowe

void		size_t
bool		ptrdiff_t
		string
byte		ubyte
short		ushort
int		uint
long		ulong
cent		ucent
char	wchar	dchar
float	ifloat	cfloat
double	idouble	cdouble
real	ireal	creal

Funkcje

<http://dpaste.dzfl.pl/96e67afe>



Struktury i klasy

- struct => semantyka wartości (kompilator może je dowolnie przesuwac, kopiowanie wywołuje this(this))
- class => semantyka referencyjna ()
- Wymuszanie atrybutów takich jak: scope, shared, const, immutable



Alias this

- <http://dpaste.dzfl.pl/fdd02a8c>



Scope

- `scope(exit)`
- `scope(success)`
- `scope(failure)`
- <http://dpaste.dzfl.pl/70cf2a13>



Final switch / override

- <http://dpaste.dzfl.pl/88e64cfc>
- <http://dpaste.dzfl.pl/19716489>



Metaprogramowanie

- <http://dpaste.dzfl.pl/8ff66400>
- CtRegex
- > Fresh timings:
- > FReD C-T 3.4 sec
- > FReD R-T 6.6 sec
- > RE2 4.8 sec
- > V8 JS 3.7 sec



Metaprogramowanie

- Static inheritance
- CTFE
- Alias typeof(obj.b) ObjB;
- Func.tupleof



Czego nie trzeba robić (1)

```
// LMAX disruptor (Java)
public long p1, p2, p3, p4, p5, p6, p7; // cache line padding
private volatile long cursor = INITIAL_CURSOR_VALUE;
public long p8, p9, p10, p11, p12, p13, p14; // cache line padding
```



Czego nie trzeba robić (2)

// Id Tech 4 (real-life C++98)

```
float idMat6::Determinant( void ) const {
```

```
// 2x2 sub-determinants required to calculate 6x6 determinant
float det2_45_01 = mat[4][0] * mat[5][1] - mat[4][1] * mat[5][0];
float det2_45_02 = mat[4][0] * mat[5][2] - mat[4][2] * mat[5][0];
float det2_45_03 = mat[4][0] * mat[5][3] - mat[4][3] * mat[5][0];
float det2_45_04 = mat[4][0] * mat[5][4] - mat[4][4] * mat[5][0];
float det2_45_05 = mat[4][0] * mat[5][5] - mat[4][5] * mat[5][0];
float det2_45_12 = mat[4][1] * mat[5][2] - mat[4][2] * mat[5][1];
float det2_45_13 = mat[4][1] * mat[5][3] - mat[4][3] * mat[5][1];
float det2_45_14 = mat[4][1] * mat[5][4] - mat[4][4] * mat[5][1];
float det2_45_15 = mat[4][1] * mat[5][5] - mat[4][5] * mat[5][1];
float det2_45_23 = mat[4][2] * mat[5][3] - mat[4][3] * mat[5][2];
float det2_45_24 = mat[4][2] * mat[5][4] - mat[4][4] * mat[5][2];
float det2_45_25 = mat[4][2] * mat[5][5] - mat[4][5] * mat[5][2];
float det2_45_34 = mat[4][3] * mat[5][4] - mat[4][4] * mat[5][3];
float det2_45_35 = mat[4][3] * mat[5][5] - mat[4][5] * mat[5][3];
float det2_45_45 = mat[4][4] * mat[5][5] - mat[4][5] * mat[5][4];
```

```
// 3x3 sub-determinants required to calculate 6x6 determinant
float det3_345_012 = mat[3][0] * det2_45_12 - mat[3][1] * det2_45_02 + mat[3][2] * det2_45_01;
float det3_345_013 = mat[3][0] * det2_45_13 - mat[3][1] * det2_45_03 + mat[3][3] * det2_45_01;
float det3_345_014 = mat[3][0] * det2_45_14 - mat[3][1] * det2_45_04 + mat[3][4] * det2_45_01;
float det3_345_015 = mat[3][0] * det2_45_15 - mat[3][1] * det2_45_05 + mat[3][5] * det2_45_01;
float det3_345_023 = mat[3][0] * det2_45_23 - mat[3][2] * det2_45_03 + mat[3][3] * det2_45_02;
float det3_345_024 = mat[3][0] * det2_45_24 - mat[3][2] * det2_45_04 + mat[3][4] * det2_45_02;
float det3_345_025 = mat[3][0] * det2_45_25 - mat[3][2] * det2_45_05 + mat[3][5] * det2_45_02;
float det3_345_034 = mat[3][0] * det2_45_34 - mat[3][3] * det2_45_04 + mat[3][4] * det2_45_03;
float det3_345_035 = mat[3][0] * det2_45_35 - mat[3][3] * det2_45_05 + mat[3][5] * det2_45_03;
float det3_345_045 = mat[3][0] * det2_45_45 - mat[3][4] * det2_45_05 + mat[3][5] * det2_45_04;
float det3_345_123 = mat[3][1] * det2_45_23 - mat[3][2] * det2_45_13 + mat[3][3] * det2_45_12;
float det3_345_124 = mat[3][1] * det2_45_24 - mat[3][2] * det2_45_14 + mat[3][4] * det2_45_12;
float det3_345_125 = mat[3][1] * det2_45_25 - mat[3][2] * det2_45_15 + mat[3][5] * det2_45_12;
float det3_345_134 = mat[3][1] * det2_45_34 - mat[3][3] * det2_45_14 + mat[3][4] * det2_45_13;
float det3_345_135 = mat[3][1] * det2_45_35 - mat[3][3] * det2_45_15 + mat[3][5] * det2_45_13;
float det3_345_145 = mat[3][1] * det2_45_45 - mat[3][4] * det2_45_15 + mat[3][5] * det2_45_14;
float det3_345_234 = mat[3][2] * det2_45_34 - mat[3][3] * det2_45_24 + mat[3][4] * det2_45_23;
float det3_345_235 = mat[3][2] * det2_45_35 - mat[3][3] * det2_45_25 + mat[3][5] * det2_45_23;
float det3_345_245 = mat[3][2] * det2_45_45 - mat[3][4] * det2_45_25 + mat[3][5] * det2_45_24;
float det3_345_345 = mat[3][3] * det2_45_45 - mat[3][4] * det2_45_35 + mat[3][5] * det2_45_34;
```

```
// 4x4 sub-determinants required to calculate 6x6 determinant
float det4_2345_0123 = mat[2][0] * det3_345_123 - mat[2][1] * det3_345_023 + mat[2][2] * det3_345_013 - mat[2][3] * det3_345_012;
float det4_2345_0124 = mat[2][0] * det3_345_124 - mat[2][1] * det3_345_024 + mat[2][2] * det3_345_014 - mat[2][4] * det3_345_012;
float det4_2345_0125 = mat[2][0] * det3_345_125 - mat[2][1] * det3_345_025 + mat[2][2] * det3_345_015 - mat[2][5] * det3_345_012;
float det4_2345_0134 = mat[2][0] * det3_345_134 - mat[2][1] * det3_345_034 + mat[2][2] * det3_345_014 - mat[2][4] * det3_345_013;
float det4_2345_0135 = mat[2][0] * det3_345_135 - mat[2][1] * det3_345_035 + mat[2][3] * det3_345_015 - mat[2][5] * det3_345_013;
float det4_2345_0145 = mat[2][0] * det3_345_145 - mat[2][1] * det3_345_045 + mat[2][4] * det3_345_015 - mat[2][5] * det3_345_014;
float det4_2345_0234 = mat[2][0] * det3_345_234 - mat[2][2] * det3_345_034 + mat[2][3] * det3_345_024 - mat[2][4] * det3_345_023;
float det4_2345_0235 = mat[2][0] * det3_345_235 - mat[2][2] * det3_345_035 + mat[2][3] * det3_345_025 - mat[2][5] * det3_345_023;
float det4_2345_0245 = mat[2][0] * det3_345_245 - mat[2][2] * det3_345_045 + mat[2][4] * det3_345_025 - mat[2][5] * det3_345_024;
float det4_2345_0345 = mat[2][0] * det3_345_345 - mat[2][3] * det3_345_045 + mat[2][4] * det3_345_035 - mat[2][5] * det3_345_034;
float det4_2345_1234 = mat[2][1] * det3_345_234 - mat[2][2] * det3_345_134 + mat[2][3] * det3_345_124 - mat[2][4] * det3_345_123;
float det4_2345_1235 = mat[2][1] * det3_345_235 - mat[2][2] * det3_345_135 + mat[2][3] * det3_345_125 - mat[2][5] * det3_345_123;
float det4_2345_1245 = mat[2][1] * det3_345_245 - mat[2][2] * det3_345_145 + mat[2][4] * det3_345_125 - mat[2][5] * det3_345_124;
float det4_2345_1345 = mat[2][1] * det3_345_345 - mat[2][3] * det3_345_145 + mat[2][4] * det3_345_135 - mat[2][5] * det3_345_134;
float det4_2345_2345 = mat[2][2] * det3_345_345 - mat[2][3] * det3_345_245 + mat[2][4] * det3_345_235 - mat[2][5] * det3_345_234;
```

```
// 5x5 sub-determinants required to calculate 6x6 determinant
```

```
float det5_12345_01234 = mat[1][0] * det4_2345_1234 - mat[1][1] * det4_2345_0234 + mat[1][2] * det4_2345_0134 - mat[1][3] * det4_2345_0124 + mat[1][4] * det4_2345_0123;
float det5_12345_01235 = mat[1][0] * det4_2345_1235 - mat[1][1] * det4_2345_0235 + mat[1][2] * det4_2345_0135 - mat[1][3] * det4_2345_0125 + mat[1][5] * det4_2345_0123;
```



Programowanie kontraktowe

- <http://dpaste.dzfl.pl/83242937>
-
-
- Dziedziczenie:
 - in musi zezwalać przynajmniej na tyle ile jego rodzice (poluzowanie)
 - out dla każdego interfejsu musi dotrzymać przynajmniej tyle co jego rodzice (zacieśnianie)
 -
- Obiekty dodatkowo mogą posiadać niezmiennik (invariant), który jest sprawdzany dla trybu debugowania:
 - po wywołaniu konstruktora
 - przed wywołaniem destruktora
 - przed wywołaniem dowolnej metody publicznej
 - po wywołaniu dowolnej metody publicznej



get / set

- <http://dpaste.dzfl.pl/2e403833>



Tablice

<http://dpaste.dzfl.pl/c3f74f1f>



Modyfikatory

- Const/immutable/shared
- Ref/in/out/inout



Moduły

- `Import std.stdio : mytext = text;`
- `Void main() { import std.stdio; }`
- `Private import std.stdio;`
- `Public import std.assert;`
- `Static this(){} static ~this(){}`



Wielowątkowość

- Standardowo - thread local
- Wyjątkowo – shared
- `class Foo { synchronized void bar() {
...statements... } }`
-
- Pure
-
- `Synchronized { }`



Mixin

- Observer pattern?



Wady

- Bugi
- Toolchain!
- Razem z możliwościami metaprogramowania - komunikaty błędów. Odpowiednia walidacja szablonów spoczywa na barkach programisty.



Dziękuję za uwagę

- <http://dlang.org/>
- <http://wiki.dlang.org/>
- <http://forum.dlang.org/>

