interface IInterface

{

void DoSomething();

}

namespace MyApplication

{

/\*

\* This ia a test class.

\*/

class SomeClass : IInterface

{

array<float> m\_arr;

array<SomeClass@> m\_children;

array<array<SomeClass@>> m\_subChildren; // Nested templates

int m\_thing;

SomeClass()

{

// Add some integers

m\_arr.insertLast(1.0f);

m\_arr.insertLast(1.75f);

m\_arr.insertLast(3.14159f);

uint x = 0x7fff0000;

int y = 9001;

}

int get\_Thing() property { return m\_thing; }

void set\_Thing(int x) property { m\_thing = x; }

void DoSomething()

{

print("Something! " + 'stuff.');

for (uint i = 0; i < m\_arr.length(); i++) {

print(" " + i + ": " + m\_arr[i]);

}

}

protected void SomeProtectedFunction()

{

try {

DoSomething();

} catch {

print("Exception while doing something!");

}

}

}

}

void Main()

{

SomeClass@ c = SomeClass();

c.DoSomething();

}