

Proposal Improvements to Gabriel dos Reis' modules

Author: José Rodrigo
E-mail: rodrigojose690@gmail.com
Date: 2016-09-10
Doc. Version: N0001

Table of Contents

I.	Introduction	1
II.	Technical Specifications	1
III.	Acknowledgements	3

I. Introduction

The actual Gabriel dos Reis' modules proposal will change the way with C++ projects are currently made, making better the development speeding up the compilation and organizing the projects. The current proposal is good, but some issues are missing and I bring in this proposal solutions for those.

II. Technical Specifications

1. Standard C++ modules in std. prefix

This solve the needed to make difference between modules into compilers paths and into user paths. Looks like `import java.util.Scanner;` of Java packages.

Look at this:

```
module Main;

import std.io; // (1)
import std.sys.time; // (1)
import window; // (2)

// ...
```

(1) Look at the standard compiler path for modules, (2) Look at user path for modules.

Obs.: for that works is needed to disabled the possibility of modules with dots in their names.

2. A *implementation* keyword

This keyword is to files that are implementing class, functions,Fox example, look at this project:

```

// Foo.h
module Foo;

import std.io;

export class Foo
{
public:
    void bar();
}

// Foo.cpp
implementation Foo;

void Foo::bar()
{
    std::cout << "Foo::bar()";
}

// Main.cpp
import Foo;

int main()
{
    Foo foo;

    foo.bar();
}

```

The *implementation* keyword make the compiler put the Foo.cpp to be compiled after Foo.h, because it is a implementation. Is better intuitive than *module partition "Foo"*;

2. Multiple entry points in a project

Allow more than one main() function in a same project, one by module, for independent module testing.

Look at this:

```

module Foo;

int main()
{
    Foo foo;

    foo.bar();
}

module Window;

export int main()
{

```

```
Window window;  
  
window.show();  
}
```

As the `main()` in the `Foo` isn't exported, that project have don't conflict of names. And, as the `main()` in the `Window` is exported, it can be used as project entry point.

III. Acknowledgements

The author wishes the Gabriel dos Reis for the great work in the Modules proposal who is evolving in each whitepaper.