using DBus;

namespace Test {

class Foo : Object {

public signal void some\_event (); // definition of the signal

public void method () {

some\_event (); // emitting the signal (callbacks get invoked)

}

}

}

/\* defining a class \*/

class Track : GLib.Object, Test.Foo { /\* subclassing 'GLib.Object' \*/

public double mass; /\* a public field \*/

public double name { get; set; } /\* a public property \*/

private bool terminated = false; /\* a private field \*/

public void terminate() { /\* a public method \*/

terminated = true;

}

}

const ALL\_UPPER\_CASE = "you should follow this convention";

var t = new Track(); // same as: Track t = new Track();

var s = "hello"; // same as: string s = "hello";

var l = new List<int>(); // same as: List<int> l = new List<int>();

var i = 10; // same as: int i = 10;

#if (ololo)

Regex regex = /foo/;

#endif

/\*

\* Entry point can be outside class

\*/

void main () {

var long\_string = """

Example of "verbatim string".

Same as in @"string" in C#

"""

var foo = new Foo ();

foo.some\_event.connect (callback\_a); // connecting the callback functions

foo.some\_event.connect (callback\_b);

foo.method ();

}