

|  |  |  |
| --- | --- | --- |
|  | C代码调用Lua函数 | |
| Brief Description | Lua的基本扩展功能之一 | |
| Precondition | 无 | |
| Primary Actor | 用户 | |
| Second Actor | 无 | |
| Dependency | 基本功能 | |
| Generalization | 无 | |
| Basic Flow | 描述基本流 | |
| Step 1 | Lua的初始化，包括创建状态和引用库等 |
| Step 2 | 解析并编译Lua代码 |
| Step 3 | 用Lua提供的函数指明要调用的Lua函数 |
| Step 4 | 用Lua提供的函数传递上一步需要的参数 |
| Step 5 | 调用Lua\_pcall调用函数 |
| Step 6 | 调用Lua的函数获得调用返回值结果 |
| Postcondition | 可以使用返回值做下一步操作 |
| Specific Alternative Flows | 无 | |
| RFS |  |
| Step 1 |  |
| Step 2 |  |
| Postcondition |  |
| Global Alternative Flows | 可能出现异常 | |
| Step 1 | 进入异常处理进程 |
| Postcondition | 会恢复到进入异常前的状态 |
| Bounded Alternative Flows | 无 | |
| RFS |  |
| Step 1 |  |
| Step 2 |  |
| Postcondition |  |

|  |  |  |
| --- | --- | --- |
| Use Case Name | Lua代码调用C函数 | |
| Brief Description | Lua的基本扩展功能之一 | |
| Precondition | 无 | |
| Primary Actor | 用户 | |
| Second Actor | 无 | |
| Dependency | 基本功能 | |
| Generalization | 无 | |
| Basic Flow | 描述基本流 | |
| Step 1 | 对要调用的函数注册 |
| Step 2 | 调用上面注册的函数 |
| Step 3 | 用Lua提供的函数获取函数参数 |
| Step 4 | 用Lua提供的函数获取函数来获取返回值 |
| Postcondition | 可以使用返回值做下一步操作 |
| Specific Alternative Flows | 无 | |
| RFS |  |
| Step 1 |  |
| Postcondition |  |
| Global Alternative Flows | 可能出现异常 | |
| Step 1 | 进入异常处理进程 |
| Postcondition | 会恢复到进入异常前的状态 |
| Bounded Alternative Flows | 无 | |
| RFS |  |
| Step 1 |  |
| Step 2 |  |
| Postcondition |  |

(所谓的动态加载无非是写成库，在需要的时候调用一下就行了。)

|  |  |  |
| --- | --- | --- |
| Use Case Name | 内嵌库使用例子（5.2版本新的需求） | |
| Brief Description | 以math库为例，介绍Lua的内嵌库的需求 | |
| Precondition | 没有库时，Lua可以正常运行 | |
| Primary Actor | 用户 | |
| Second Actor | 无 | |
| Dependency | 该用例与其他的Lua用例独立 | |
| Generalization | 无 | |
| Basic Flow | 函数调用 | |
| Step 1 | 引用math库lmathlib.c |
| Step 2 | 调用相应的库函数 |
| Postcondition | 调用结果可以被其他Lua代码引用 |
| Specific Alternative Flows | 常量调用 | |
| RFS | 1 |
| Step 1 | 引用相应的常数变量 |
| Postcondition | 调用结果可以被其他Lua代码引用 |
| Global Alternative Flows | 可能出现异常 | |
| Step 1 | 进入异常处理进程 |
| Postcondition | 会恢复到进入异常前的状态 |
| Bounded Alternative Flows | 无 | |
| RFS |  |
| Step 1 |  |
| Step 2 |  |
| Postcondition |  |

|  |  |  |
| --- | --- | --- |
| Use Case Name | 独立解析和字节码编译 | |
| Brief Description | Lua作为一种独立语言，提供的对源代码的解析和编译 | |
| Precondition | 源代码 | |
| Primary Actor | 系统 | |
| Second Actor | 用户 | |
| Dependency | 与其他编译器的解析和编译过程独立 | |
| Generalization | 虚拟机可以使用 | |
| Basic Flow | 描述基本流 | |
| Step 1 | 解释源代码，生成中间代码 |
| Step 2 | 编译源代码，生成字节码 |
|  |  |
|  |  |
| Postcondition | 字节码可以提供给虚拟机使用 |
| Specific Alternative Flows | 无 | |
| RFS |  |
| Step 1 |  |
| Postcondition |  |
| Global Alternative Flows | 可能出现异常 | |
| Step 1 | 进入异常处理进程 |
| Postcondition | 会恢复到进入异常前的状态 |
| Bounded Alternative Flows | 无 | |
| RFS |  |
| Step 1 |  |
| Step 2 |  |
| Postcondition |  |

Add 异常处理进程