

# 单元测试工具 AUnit

## 使用手册

V1.0

北京轩宇信息技术有限公司

2016.03

## AUnit 使用手册变更记录

编号	修改人	修改日期	版本	主要修改内容	备注
1	王柏	2016/03/15	V1.0	新建	首次编辑
2					
3					
4					
5					
6					
7					
8					
9					
10					

# 1 目录

1	目录 .....	3
1	<b>AUNIT 产品介绍 .....</b>	<b>5</b>
1.1	表格驱动的可视化用例设计视图 .....	5
1.2	测试脚本和驱动的自动生成 .....	6
1.3	测试用例的自动生成 .....	6
1.4	在高可信度的环境中运行测试 .....	8
1.5	丰富的函数桩控制特性 .....	8
1.6	无需编译的测试用例执行 .....	8
1.7	支持覆盖率的非干预测量 .....	9
1.8	细致入微的易用性提升 .....	9
1.9	集成静态分析 .....	9
1.10	可定制的报告生成 .....	10
2	<b>AUNIT 安装 .....</b>	<b>10</b>
3	<b>快速入门 .....</b>	<b>10</b>
3.1	产品入门 .....	10
3.2	产品界面及布局 .....	11
3.2.1	菜单栏 .....	12
3.2.2	工具栏 .....	13
3.2.3	右键快捷菜单 .....	13
3.3	产品使用 .....	13
3.2.1	第一步：新建测试工程 .....	14
3.2.2	第二步：设计测试用例 .....	16
3.2.3	第三步：运行单元测试 .....	17
3.2.4	第四步：查看单元测试结果 .....	17
4	<b>测试用例设计 .....</b>	<b>18</b>

4.1	自动生成测试用例.....	18
4.2	手动设计测试用例.....	18
4.2.1	切换平铺模式.....	19
4.2.2	编辑全局变量.....	19
4.2.3	编辑用户自定义变量.....	20
4.2.4	为指针变量赋值.....	24
4.2.5	编辑和设置桩函数.....	27
4.2.6	设置输入值和输出预期.....	30
4.2.7	复制测试用例.....	31
4.2.8	源码浏览视图辅助用例设计.....	32
5	<b>运行单元测试及结果查看.....</b>	<b>33</b>
5.1	运行单元测试.....	33
5.2	查看测试结果.....	34
5.2.1	查看函数测试状态.....	34
5.2.2	查看用例执行状态.....	35
5.2.3	查看函数覆盖率.....	36
5.2.4	查看预期失败用例.....	38
5.3	生成并查看测试报告.....	39
6	<b>工程导入导出 .....</b>	<b>41</b>
6.1	导入工程.....	41
6.2	导出工程.....	41
7	<b>产品更新.....</b>	<b>41</b>
8	<b>AUNIT 支持的环境 .....</b>	<b>41</b>
9	<b>常见问题汇总 .....</b>	<b>42</b>

## 1 AUnit 产品介绍

SunwiseAUnit 是北京轩宇信息技术有限公司自主研发的、针对 C 语言程序的可视化、自动化单元测试平台。旨在通过提供一个颠覆传统的、简单易用、高度自动化的平台提高单元测试的效率，让软件开发能够在早期进行充分的模块级测试。

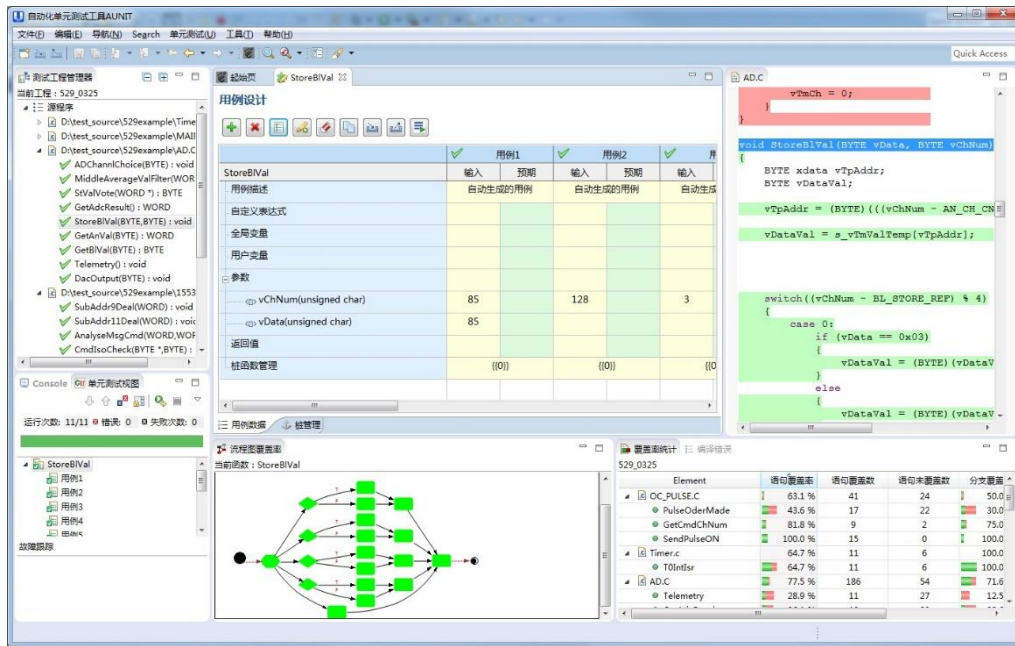


图 1 SunwiseAUnit

AUnit 的关键特性包括：提供表格驱动的可视化用例设计视图；支持测试脚本和驱动的自动生成；支持给定覆盖率的测试用例的自动生成；保证测试运行在高可信度的环境中；提供了丰富的函数桩控制特性；测试用例执行无需编译；支持以非干预方式进行覆盖率测量；细致入微的易用性设计；可选择集成静态分析产品；提供可定制的测试报告生成。

### 1.1 表格驱动的可视化用例设计视图

AUnit 为用户提供最易用的表格驱动设计视图，利用代码分析技术自动获取展示函数的接口、全局变量、底层调用等信息。用户无需编写任何代码，仅需要在表格中填充相应的输入值和返回值，就可以完成用例的设计。

用例设计	用例1		用例2		用例3	
SubAddr9Deal	输入	预期	输入	预期	输入	预期
用例描述	自动生成的用例		自动生成的用例			
自定义表达式						
XBYTE[BU_A_SACW_RA9]	0	0		1		12
全局变量						
用户变量						
user_var0(WORD [2])						
user_var0[0](WORD)	0x55		0x55		0x55	
user_var0[1](WORD)	34		34		3	
参数						
wSa9DataPtr(WORD)	21845		32		32	
返回值						
桩函数管理	{(0)}		{(0)}		{(0)}	
Buf1ToBuf2						
PulseOrderMade						
CmdIsoCheck						
pBuf(BYTE *)		NULL		NULL		NULL
vChkLth(BYTE)		0x55		0x55		0x55
returnValue(BOOL)	0x55		0x55		0x55	

图 2 表格驱动的可视化用例设计

手动设计用例更加方便快捷。用例编辑器中的测试输入默认以树状结构展示，设计用例时根据需要可以切换到平铺模式，这样所有输入项即一目了然。编辑器提供了新建用例、删除用例、复制用例、编辑全局变量、编辑用户变量、导入导出测试数据和一键构建运行测试用例等主要功能。测试用例运行之后，在编辑其中会以图标的方式显示标记出各个用例执行情况（如成功、失败、错误），如果失败，还会在失败的输入项对应的预期单元格中以红色高亮方式标记出该单元格，并反馈回实际值。若有函数调用，编辑器会在桩函数管理中默认生成一个空的桩，用户可以在此基础上重新编辑桩函数。

1.2 测试脚本和驱动的自动生成

基于可视化用例设计的结果，AUnit 为用户自动静默地生成测试执行需要的脚本和测试驱动程序。这项生成工作对用户完全透明，大大减少了开展测试所需的额外工作量。让测试人员专注于用例的设计。

1.3 测试用例的自动生成

基于最前沿的动态符号执行技术和约束求解技术，AUnit 能够支持大多数 C 程序测试用例的快速生成，所生成的用例无需任何改动即可执行，并能够满足几

乎 100%的语句和分支覆盖要求。用例自动生成主要包含前端、单元测试用例生成引擎、混合内存模型等三个主要部分。

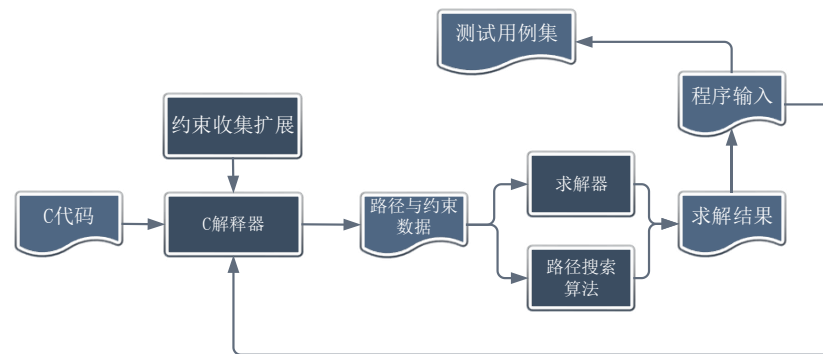


图 3 用例生成引擎

- (1) 前端处理器（FrontEnd）：给定源代码和编译配置，对被测软件代码进行预处理、词法/语法分析、CFG 构建和符号表构建，产生后续分析所需要的各种中间表示 IR。
- (2) 单元测试用例生成引擎（Core）：该部分是 SunwiseAUnit 的核心部分之一，以前端处理器产生的中间表示为输入，采用混合符号执行技术、约束求解技术来产生满足覆盖准则的测试输入，经过若干次迭代，最终生成测试输入集。本部分具体包括：
  - a) C 源码级解释器（CInt）：对给定中间表示进行具体解释执行，根据控制流图对具体内存模型进行存取，并完成函数调用、条件跳转等控制流的执行；
  - b) 符号执行器（SymEx）：满足真实嵌入式 C 程序，与源码级解释器同步执行，在具体执行过程中，维护抽象内存模型的状态；
  - c) 约束收集器（ConsGen）：在符号执行引导下，收集一条路径的约束公式；
  - d) 约束求解器（Solver）：对收集的布尔约束进行求解，AUnit 采用 Z3 求解器；
  - e) 高效和高覆盖率的路径搜索及选择策略（PathSel）：根据生成的输入以及历史数据，采用高效的搜索算法，确定下一个要执行的路径，并指导解释器迭代执行。

- (3) 混合内存模型 (MemModel)：混合内存模型是解释执行和符号执行的基础，提供对 C 语言各类语言结构、类型、运算的内存模型支持；具体地，又分为具体内存模型和抽象内存模型。

AUnit 在对 C 源码进行解释执行时，支持基本的 C 语言语法特性和特定异常检查，具体为：

- 1) 支持类型强转，包括显式强转和隐式强转；
- 2) 支持函数指针调用，指针运算等操作；
- 3) 支持 Mock 函数，对于没有定义的函数，通过 Mock 的方法完成函数调用；
- 4) 支持内存越界访问检测；
- 5) 支持对内存改变及其他动作的监听；
- 6) 支持默认初始化变量，默认初始化值为 0；
- 7) 其他基本语法支持。

## 1.4 在高可信度的环境中运行测试

AUnit 的测试执行环境采用自主研发的指令集模拟器，无需依赖第三方的调试器或模拟器，也无需连接硬件目标板，就能实现在与目标机一致的环境中进行测试。

## 1.5 丰富的函数桩控制特性

AUnit 支持多样化、灵活的函数桩机制，用于隔离测试。默认情况下，AUnit 会自动分析出被测函数调用的底层模块及其接口，并生成空的函数桩，以便用户可以快速地开展测试。函数桩提供表格式的实例化方式，支持函数调用序列验证、函数实参验证、全局变量副作用指定、多次调用指定不同函数桩等丰富特性。

## 1.6 无需编译的测试用例执行

AUnit 提供无需编译的测试用例执行模式，用户无需依赖复杂编译环境和设置，仅使用编译好的被测软件就可以进行单元测试。在该模式下，测试用例数据自动转换为目标虚拟机的驱动脚本，通过这些脚本完成测试驱动、测试用例的输



入（包括桩）设置以及预期结果比较。这大大减少了测试过程中由于频繁编译带来的开销。

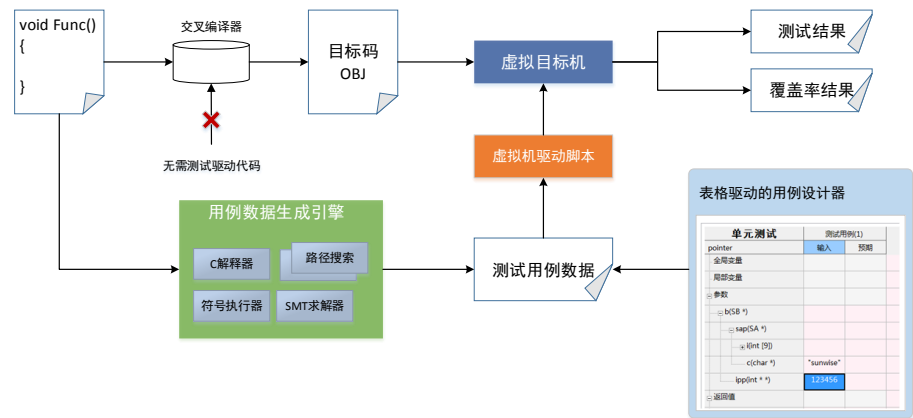


图 4 无需编译的测试用例执行

### 1.7 支持覆盖率的非干预测量

利用模拟器自身提供的覆盖率监控技术，AUnit 能够无需对代码进行插桩就获得基于源代码和目标码的语句、分析、MC/DC 覆盖率信息，支持单个用例或用例集的覆盖率叠加。

AUnit 提供基于代码文本标注和流程图标注的覆盖率视图，最大限度地为用户提供可视化的覆盖率信息。

### 1.8 细致入微的易用性提升

AUnit 从多个方面对工具的易用性进行了提升，让用户专注于用例设计。易用性特性包括：大量表格操作的快捷键、覆盖率信息的可视化即时显示、测试异常结果高亮、统一的操作台等等。

在实际工程实测中，一个经过简单培训的测试人员使用 AUnit 对 3000 行代码进行人工测试，仅需要 4 小时。

### 1.9 集成静态分析

AUnit 可选集成轩宇信息公司的静态分析产品 SpecChecker，用户可在统一的界面下在开发早期就完成两项工作。静态分析的功能包括编码标准检查、中断数据竞争分析、代码度量等。



图 5 SpecChecker

### 1.10 可定制化的测试报告生成

AUnit 可生成满足归档要求（如符合 GJB-141）的单元测试报告，支持格式包括 HTML、WORD、PDF 等。同时，报告格式支持可定制。

## 2 AUnit 安装

暂无。

## 3 快速入门

运行安装目录下的 AUnit 可执行程序，启动 AUnit 图形用户界面。第一次启动时，工具会提示选择工作区路径。此后在 AUnit 中创建的测试工程将都存放在该工作区中，建议用户选择固定目录作为工作区。

### 3.1 产品入门

首次启动 AUnit，可以看到产品起始页。起始页左侧部分提供了新建项目、打开项目、打开示例项目、打开帮助文档以及打开最近编辑项目的链接；右侧

部分提供了 AUnit 产品快速入门、使用指南及视频教学动画演示的快捷方式，如图 6 所示。



图 6 AUnit 工具起始页

在起始页左下角有“加载项目后关闭起始页”和“启动时打开起始页”两个复选框，用户可根据需要勾选相关选项。

3.2 产品界面及布局

关闭起始页后进入 AUnit 主界面，如图 7 所示。

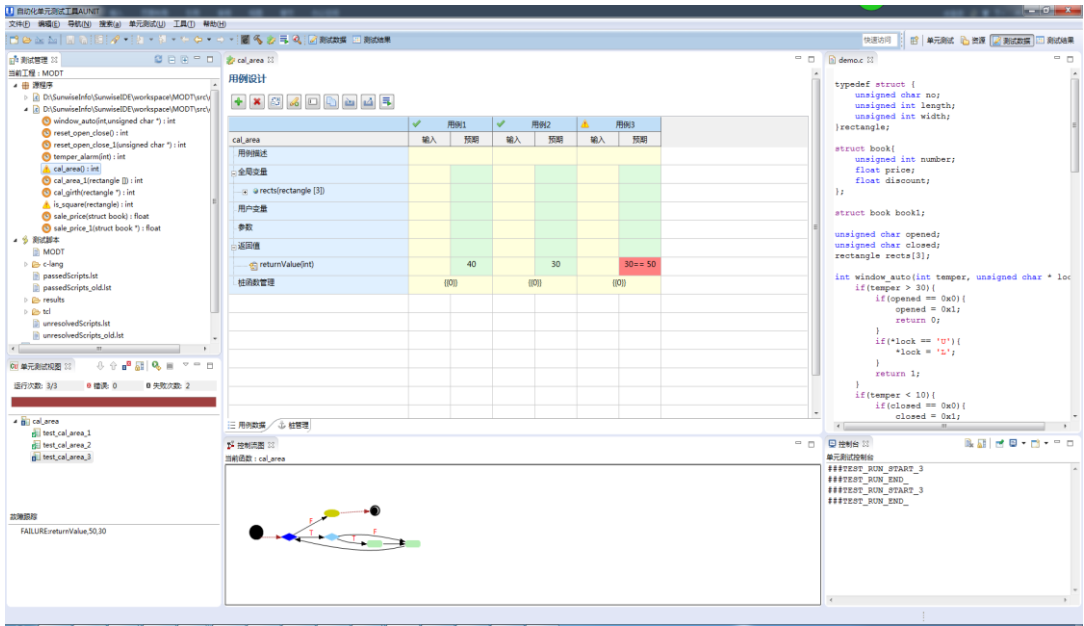


图 7 AUnit 主界面

- 1) 测试管理视图：显示当前加载的测试工程，可以将各目录展开查看源码及测试脚本等详细信息。

- 2) **用例设计编辑器**：SunwiseAUnit 提供了网格化用例设计编辑器。将测试管理视图中的源程序展开到函数一级，双击任意函数打开用例设计编辑器，即可开始编辑当前函数的测试用例。
- 3) **单元测试视图**：该视图显示运行单元测试过程中用例执行情况，包括用例总数，错误、失败次数等信息。
- 4) **源代码及覆盖信息查看器**：显示当前被测函数的源程序的内容。在用例设计过程中可以辅助用例设计，用例执行后会以高亮方式显示被测函数的覆盖率信息。
- 5) **覆盖率视图**：显示当前被测函数及其所在的源文件和工程的测试覆盖信息，包括详细的语句及分支覆盖信息。
- 6) **控制流覆盖视图**：该视图以结构化图形的方式展示当前被测函数的语句及分支覆盖情况，更加直观。
- 7) **菜单栏**：单元测试菜单包含自动生成测试用例、生成用例脚本、运行测试用例、生成测试报告和配置高级运行等功能。
- 8) **工具栏**：包含新建、打开、导入、导出测试工程，打开起始页按钮，重新构建工程、运行测试用例、打开单元测试报告、测试数据和测试结果等按钮。
- 9) **右键弹出菜单**：在源程序中选中某个待测函数后，点击右键弹出的菜单，包含了与菜单栏单元测试菜单相同的功能。

### 3.2.1 菜单栏

AUnit 的菜单栏中常用的菜单有单元测试(U)、工具(T)等，如图 8 所示。

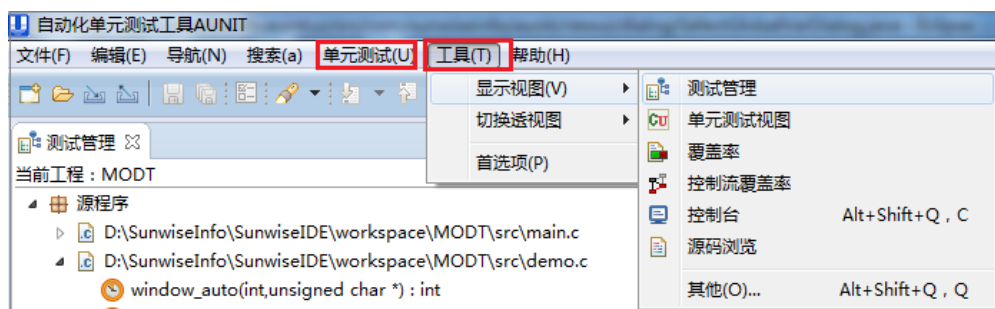


图 8 菜单栏

### 3.2.2 工具栏

AUnit 的工具栏功能与菜单栏功能相似，工具栏提供了更加方便快捷的方式来使用本工具。如图 9 所示：



图 9 工具栏

工具栏中图标与菜单栏中图标是相互对应的，对应图标的功能详细请查询菜单栏一节即可。

### 3.2.3 右键快捷菜单

在测试管理视图中，选中某个待测源程序并展开到函数级，选中待测函数，点击右键，即可打开右键菜单。右键弹出菜单的功能与菜单栏、工具栏是一致的。如图 10 所示：

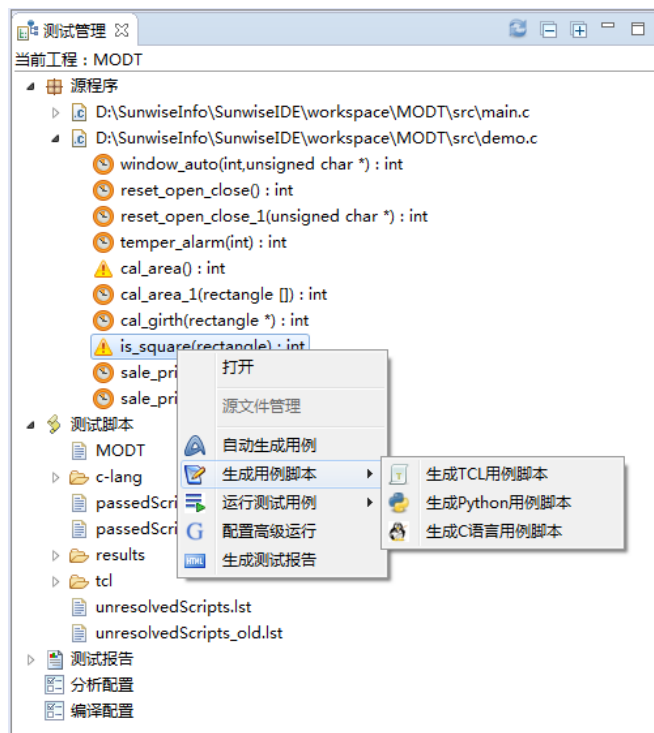


图 10 右键快捷菜单


## 3.3 产品使用

启动 AUnit 后，首先创建一个测试工程，然后打开某个待测函数的用例编辑

器，设计好测试用例后，生成测试用例脚本，然后运行测试用例。

### 3.2.1 第一步：新建测试工程

在做单元测试相关工作之前，首先要在 AUnit 工具中，通过工程向导创建一个 C 程序的测试工程，向导会要求用户输入测试工程名、选择工程创建方式（如选择待分析的源文件目录）、选择编译器配置等。

1. 在菜单栏中选择**文件->新建**（或者在工具栏使用新建工程按钮）。
2. 出现如图 11 所示的新建工程对话框，选择 **AUNIT 工具->AUnit 测试工程**，点击下一步。

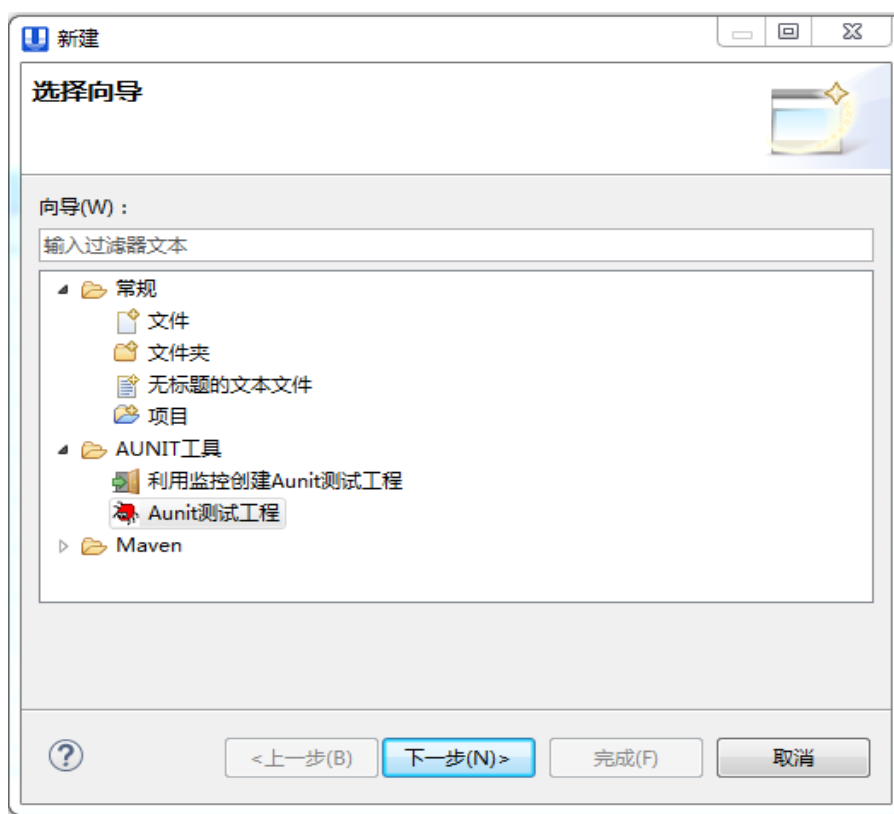


图 11 新建 AUnit 项目

3. 输入工程名称（不能与当前工作空间中的其他工程同名，不区分大小写），指定工程目录（一般默认即可）。用户可通过两种方式指定软件项目：
  - 搜索源文件目录：浏览并选择软件源代码所在目录，这种方式下，工具将能自动获取被分析软件的源文件、头文件搜索路径等其他编译选项。

- 转换外部工程文件：用户选择已有的软件工程文件（目前支持 Keil、Code Composer Studio 和 Makefile），工具将自动分析导入编译配置。如果选择已有的 Makefile 文件作为源文件，会出现“自定义 Make 命令”的复选框，选中复选框，可以修改 make 命令的参数。

选择了相应的创建方式后，选择编译器类型或者选择自定义的编译器配置文件，默认支持的编译器类型包括：ANSI、ARM、C51、DSP 系列、GNUC、IAR\_78、IAR\_80、MSP430 等。

点击**完成**按钮，即完成测试工程创建，如图 12 所示。

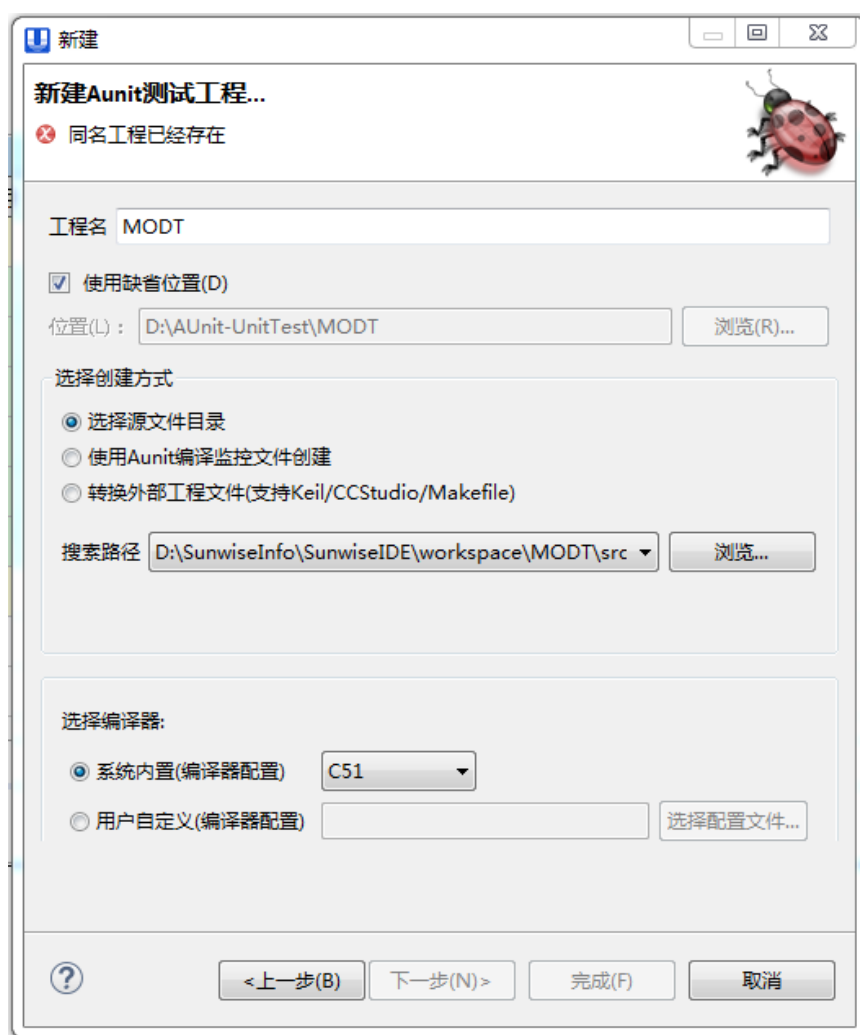


图 12 新建 AUnit 测试工程

4. 在起始页**最近**打开工程中将出现刚创建的测试工程，在测试管理视图中可看到刚刚新建的测试工程的结构，如图 13 示。

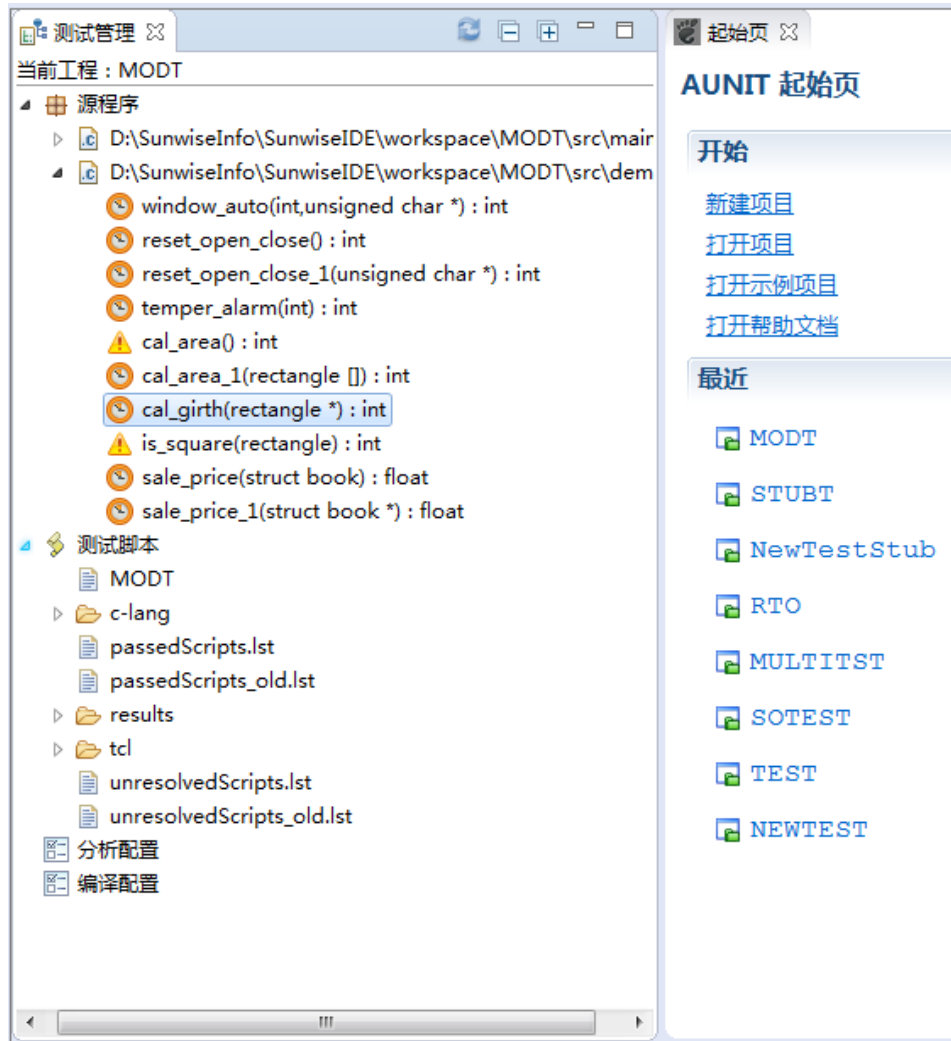










图 13 AUnit 测试工程结构

### 3.2.2 第二步：设计测试用例

在测试管理视图的**源程序**中选中某个待测源文件并展开到函数一级，选中一个待测函数，双击即可将当前函数加载到用例设计编辑器中，AUnit 提供了网格化的用例设计编辑器，如图 14 所示，点击编辑器上方的（增加测试用例）按钮，就会新增一个用例，用户只需按测试需要依次填写相应的测试输入和输出预期，即完成一个用例设计。若要添加多个用例，可选中其中已经设计好的某个用例，点击上方的按钮，复制用例，复制后再修改相应的输入和输出。选中一个用例，点击按钮，删除用例。点击按钮，可将编辑器切换到平铺模式，用例的输入输出将更加一目了然。点击按钮可新增用户变量。点击按钮可清楚所有用户自定义变量。点击和按钮，可导入、导出用例数据。



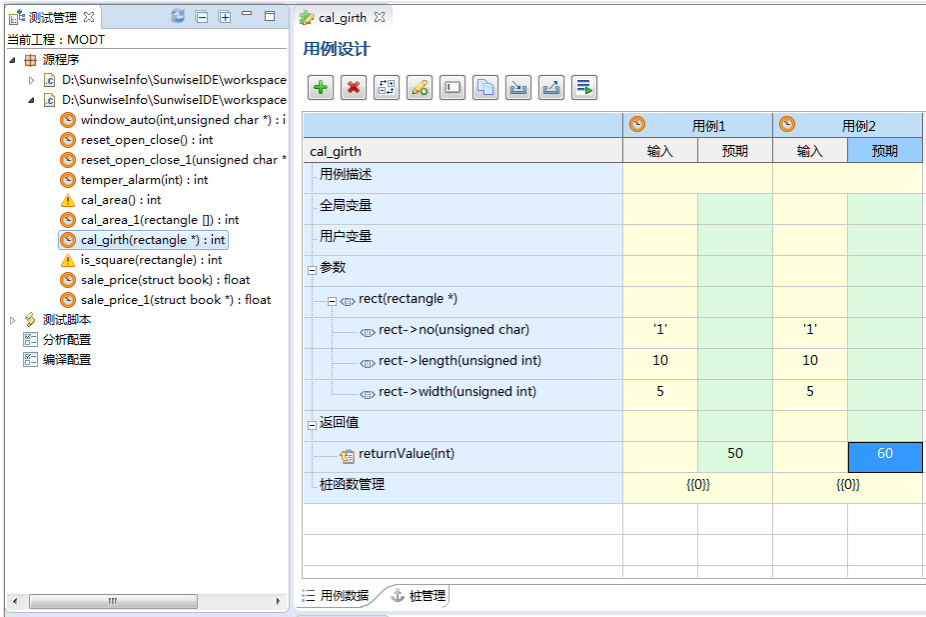



图 14 设计测试用例

3.2.3 第三步：运行单元测试

设计好用例后，共有三种途径来运行测试用例。第一种，点击编辑器上方的按钮，即可一次完成构建并运行测试用例；第二种，右击测试管理视图中的当前被测函数，在右键快捷菜单中先选择生成用例脚本->生成 C 语言用例脚本，生成测试脚本后，再选择运行测试用例->C 语言用例运行测试用例。

3.2.4 第四步：查看单元测试结果

运行完测试用例后，便可查看测试结果，此处涉及三个视图，分别是：单元测试视图、覆盖率视图和控制流覆盖视图。**单元测试视图**反映了测试用例执行情况，比如运行的用例总数、运行错误次数、失败次数以及运行错误的故障跟踪信息等；**覆盖率视图**反映的是当前被测函数以及其所在的源文件和工程的测试覆盖率信息，包括语句覆盖数、语句未覆盖数、语句覆盖率、分支覆盖数、分支未覆盖数、分支覆盖率等详细信息；**控制流覆盖视图**则以结构化图形的方式展示当前被测函数的语句及分支覆盖情况，更加直观。如图 15 所示。

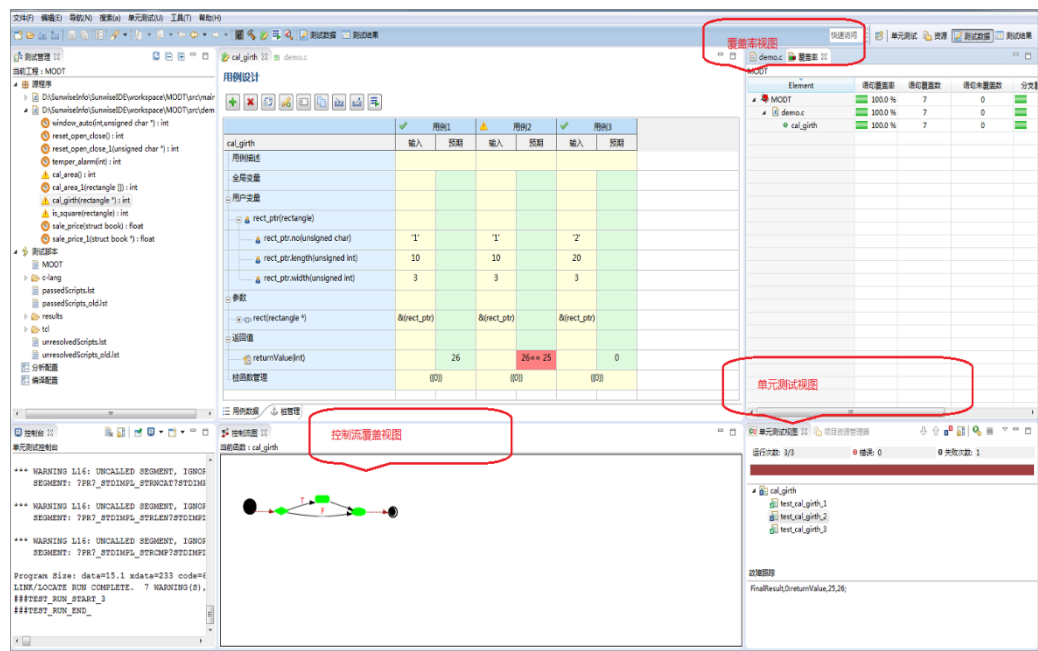



图 15 查看测试结果

## 4 测试用例设计

AUnit 同时支持自动用例生成和手动用例设计。自动生成的测试用例数据会自动匹配到用例设计编辑器中，如果有函数调用，编辑器会默认生成一个空桩，用户可以根据实际需要进行补充编辑。

### 4.1 自动生成测试用例


打开或新建一个 AUnit 测试工程后，在测试管理视图中，将**源程序**展开到函数一级，选择一个待测函数。在自动生成测试用例之前，先单击工具栏的（重新构建工程）按钮构建工程，构建完成后，在右键快捷菜单中点击“自动生成用例”选项，即可完成用例自动生成。此时双击打开被测函数用例编辑器，会看到刚生成的测试用例数据已经匹配到编辑器中，用户可根据实际需要对测试数据进行有选择地编辑，然后即可运行测试用例（参照 3.3.3 中提到的三种方式）。

### 4.2 手动设计测试用例

打开或新建一个 AUnit 测试工程后，在测试管理视图中，将**源程序**展开到函

数一级，选择一个待测函数并双击打开用例设计编辑器即可开始用例设计。在用例编辑器中可通过增删全局变量、增删用户自定义变量、指针赋值、添加桩函数、复制测试用例等主要操作来完成用例设计。

### 4.2.1 切换平铺模式

用例设计编辑器中函数的测试输入项是以树形结构展示的，虽能够清晰展示出各种数据类型的层次结果，但是当遇到复杂类型的深层嵌套时，展开时层次太过繁琐，从而显得不太方便，此时可以点击编辑器上方的  按钮，将编辑器切换到平铺模式，切换后层次结构被过滤掉，所有的输入项都一目了然，如图 16 所示。

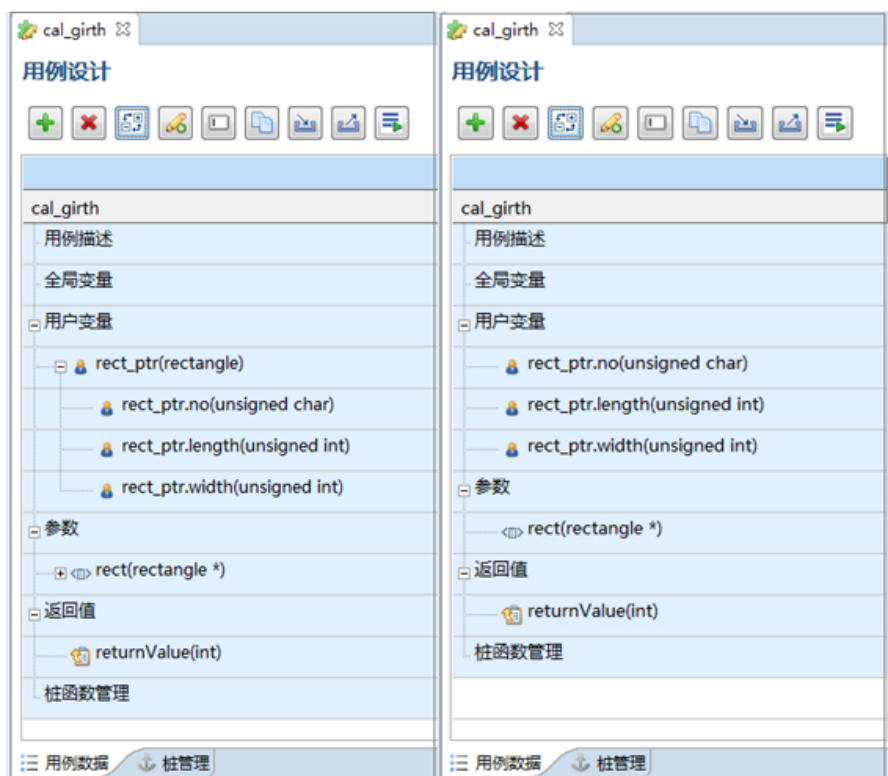







图 16 树形展示和平铺模式展示

### 4.2.2 编辑全局变量

在树形展示区域右击弹出右键快捷菜单，选择“编辑全局变量选项”，会弹出编辑全局变量对话框，如图 17 所示，具体操作如下：

- 勾选“只显示直接调用的全局变量”复选框，左侧区域将只显示当前函

数中调用的全局变量，否则显示全部全局变量；

- 双击左侧的任意一个全局变量，变量将加到右侧已选择区域，反之，双击右侧区域的某个变量，则将已选择的变量移到左侧区域；
- 选中左侧区域某个变量，点击  按钮，该变量移到右侧已选择区域，反之，选择右侧某个变量，点击  按钮，将其从已选区域中删除；
- 点击  按钮，将所有变量加到右侧已选择区域，反之，点击  按钮，将所有已选变量移到左侧区域；
- 点击  按钮，清除搜索历史，将变量顺序重置到初始状态；

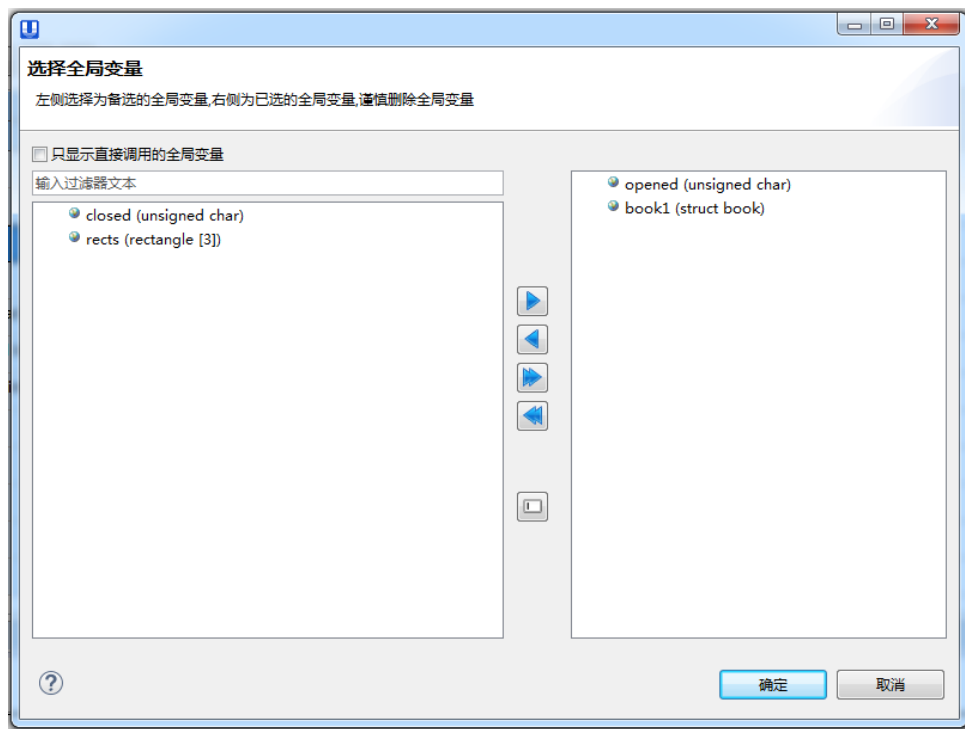



图 17 编辑全局变量

点击**确定**按钮，选择的全局变量就会加入到编辑器的“全局变量”下面。

### 4.2.3 编辑用户自定义变量

设计用例时，遇到不论是参数、全局变量还是已创建的用户自定义变量是指针的情况，且指针不指向任何全局变量，就需要用户创建一个用户变量来让其指向。有两种方式编辑用户自定义变量，分别是通过编辑区上方的用户变量编辑按钮和编辑区右键快捷菜单编辑用户变量，具体操作如下：

- **新建用户变量：** 点击  按钮，或者右击编辑器任意区域，在右键快捷菜单中选择“新建用户变量”，就会弹出新建用户变量对话框，支持的数据类型包括：基本类型、复合类型、枚举类型和自定义类型，支持的变量类型包括普通变量、数组变量和指针变量；
- 1) **新建普通变量：** 填写变量名，在变量类型中选择类型，比如选择一个自定义类型 `rectangle`，如图 18 所示，点击确定即可创建一个自定义类型的用户变量；

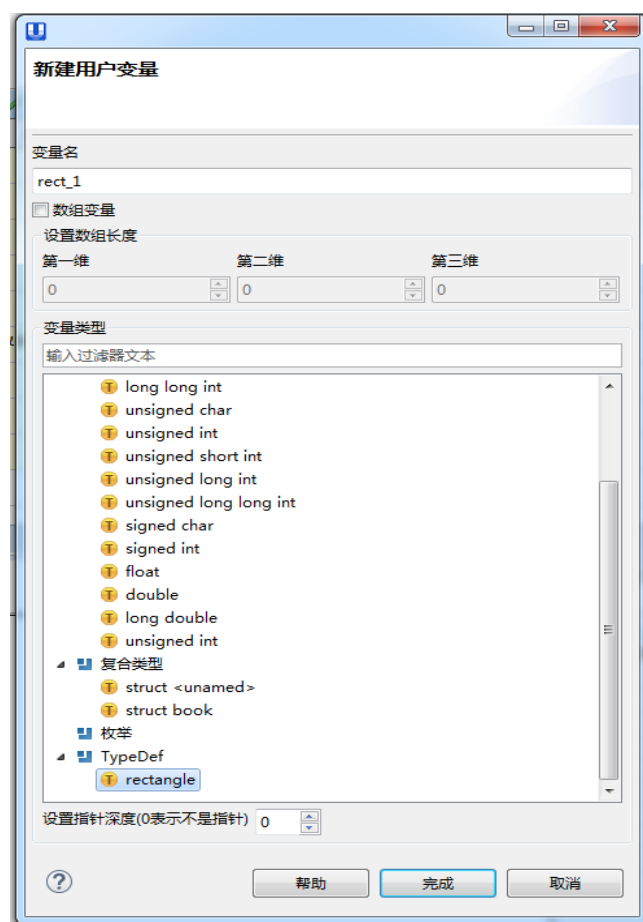


图 18 新建普通用户变量

- 2) **新建数组变量：** 如果要创建一个数组变量，与创建普通变量的区别是，需要勾选“数组变量”复选框，这样就可以激活维度设定区域来指定数组维度（目前 SunwiseAUnit 对数组维度最高支持到三维）。比如要创建一个 `int` 类型的二位数组，如图 19 所示，点击**确定**按钮，在编辑器的“用户变量”下就会出现一个变量 `uvar[3][5]`；

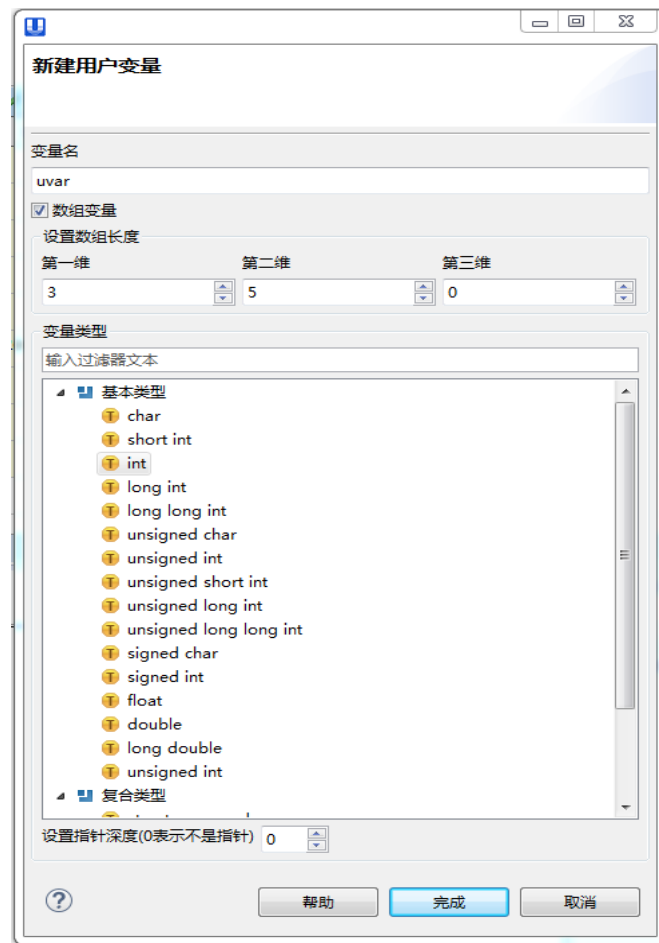


图 19 创建数组变量

- 3) **新建指针变量：**新建指针变量与新建普通变量的区别就在于是否在最下方设置了指针深度，默认 0 表示不是指针变量，大于 0 则是指针变量，数字表示指针深度（或级别）。比如创建一个自定义类型 `rectangle` 的一级指针变量，如图 20 所示；

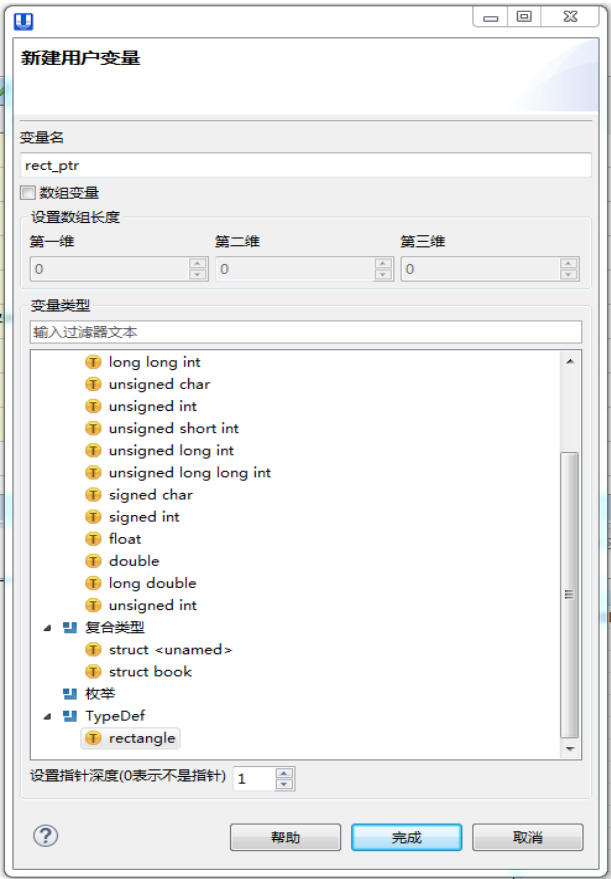


图 20 新建指针变量

用户变量创建后，在用例编辑器中显示效果如图 21 所示；

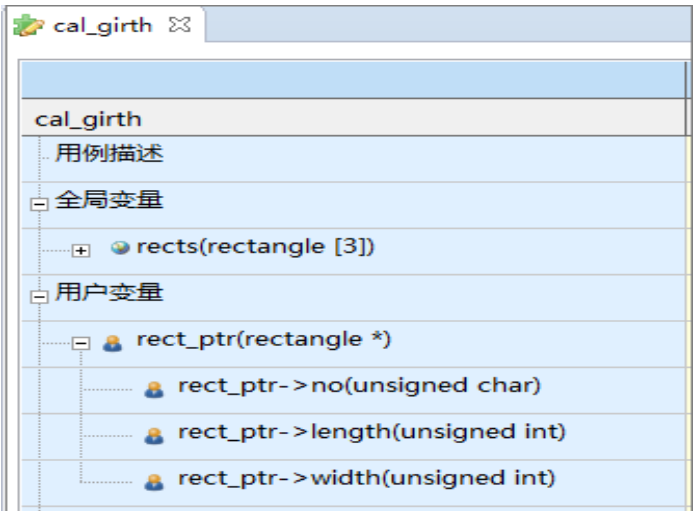



图 21 新建的用户变量

- **删除用户变量：**删除有两种方式。第一种，在编辑器中选中某个已存在的用户变量，右击弹出快捷菜单，选择删除用户变量选项，即可删除当前选中的用户变量，该方式适用于删除指定的单个用户变量，如图 22 所示；第二种，点击编辑器上方  按钮，可一次清除所有用户自定义

变量。

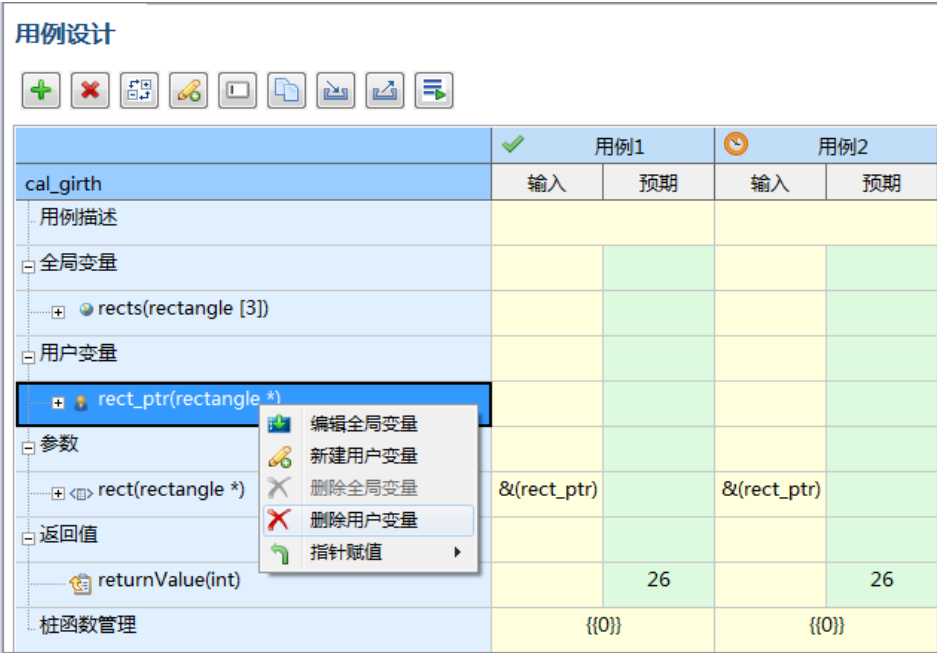


图 22 删除指定用户变量

4.2.4 为指针变量赋值

当测试输入（可能是全局变量、函数参数或已创建的用户变量）为指针变量时，为其赋值就是让其指向某个全局变量或用户变量。具体操作为：选中要



赋值的指针变量，右击弹出快捷菜单，选择**指针赋值**菜单项，如图 23 所示。

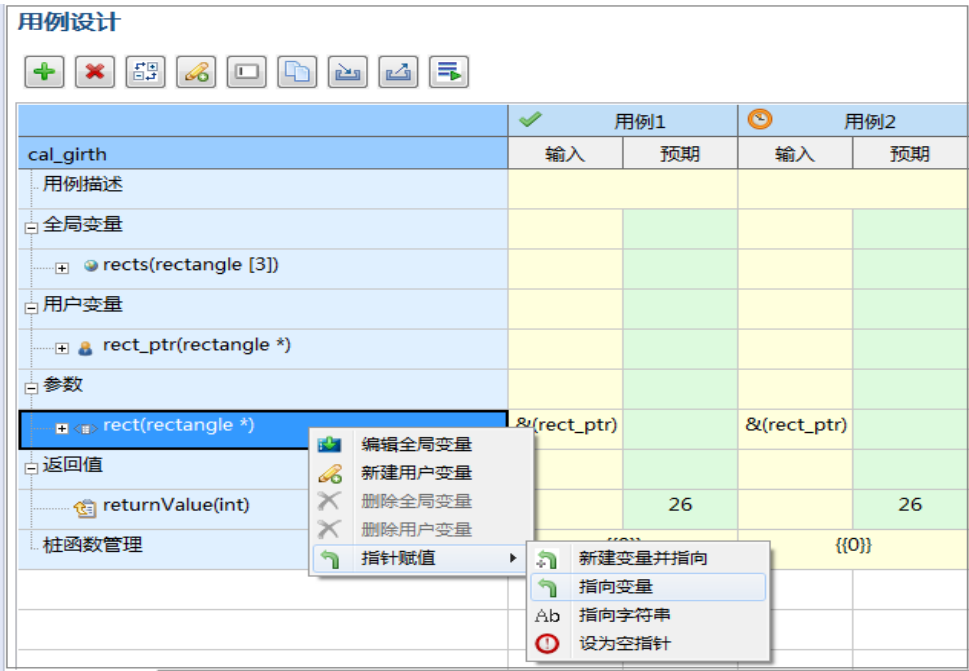


图 23 为指针变量赋值

其下包含四个子菜单项，分别为新建变量并指向、指向变量、指向字符串和设为空指针，各选项的适用场景如下（以函数参数指针为例）：

- **新建变量并指向**：该选项适用于指向一个尚未创建的用户变量。点击该选项，会先弹出新建用户变量对话框（图 18 所示），用户可根据指针数据类型创建一个合适的用户变量，点击**确定**，当前指针变量就会指向该新建的用户变量，如图 24 所示，会将新建变量地址赋给当前指针。

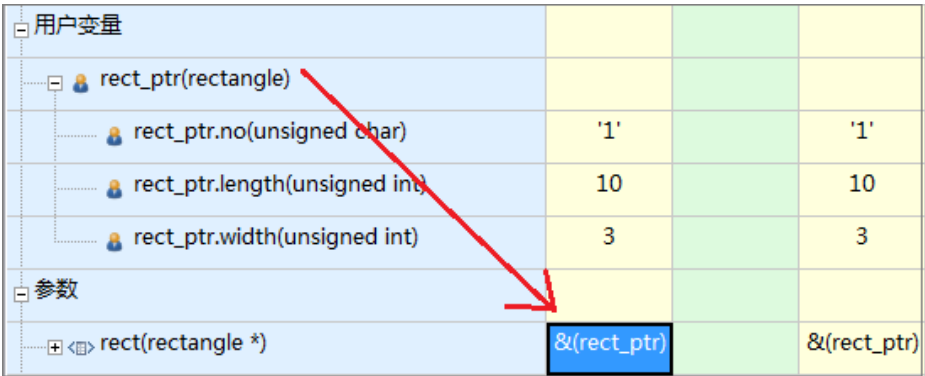


图 24 新建用户变量并指向

- **指向变量**：该选项适用于指向某个全局变量或已存在的用户变量。点击会弹出变量选择对话框，选择需要指向的变量即可，如图 25 所示。

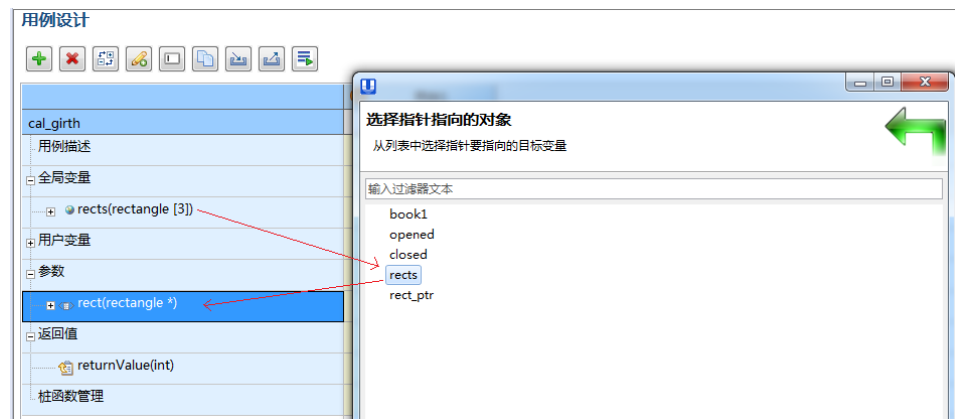


图 25 指向变量

**指向字符串：**该选项适用于字符类型指针指向一个字符串常量。此处要注意两种情况：第一种，如果想让该指针变量在所有用例中都指向同一字符串常量，那么先选中当前指针变量，再选中右键菜单中的**指向字符串**选项，会弹出“指定字符串”对话框，输入字符串并**确定**（注意，字符串要加双引号），该指针将指向该字符串常量，而且之后创建的所有用例中该指针都将指向该默认字符串，如图 26 所示。

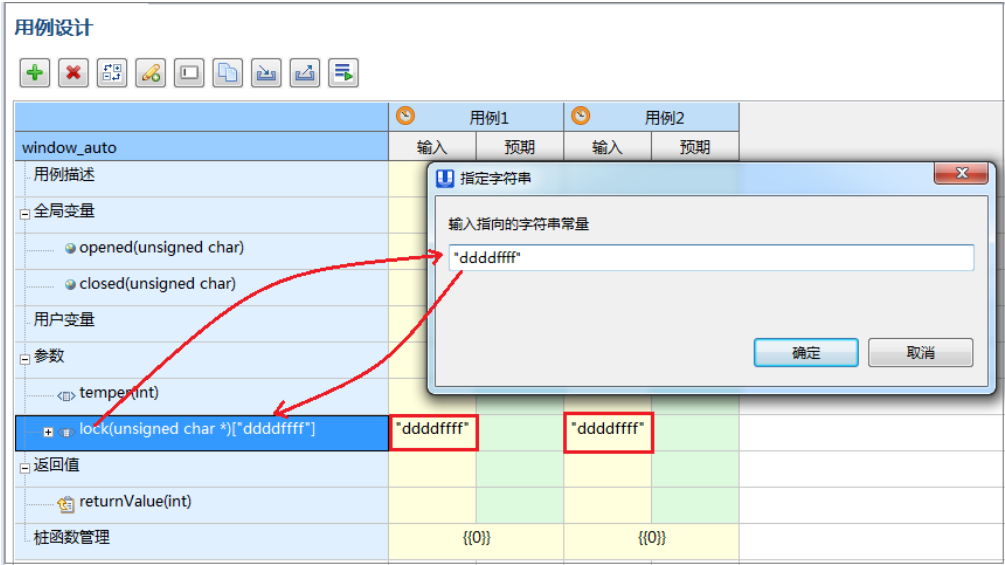


图 26 指向字符串-1

让该指针在不同用例中指向不同字符串常量，则要先选中指针变量在  
用例中的对应输入区域，后续操作同第一种，如图 27 所示。

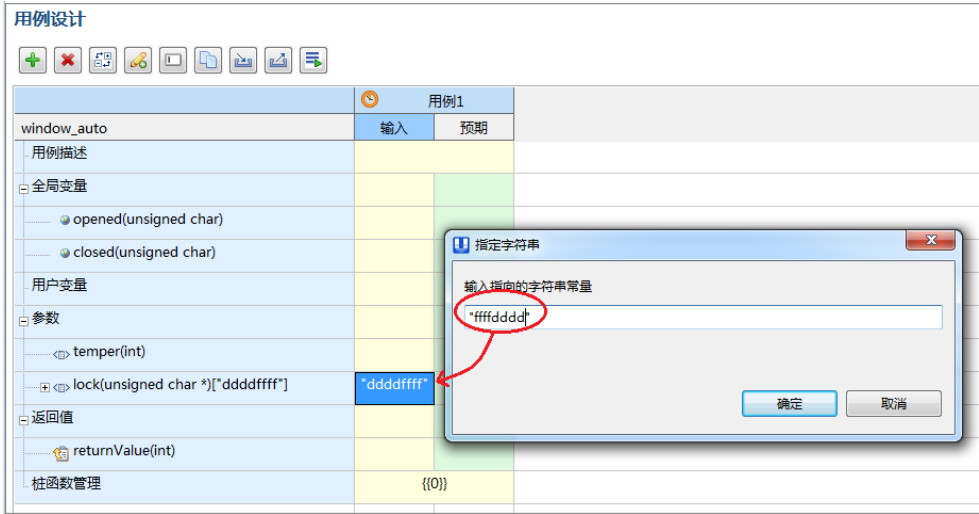


图 27 指向字符串-2

- **设为空指针：**如果想将当前指针设为空指针，只需选择右键菜单中的**设为空指针**选项，此处也分为两种情况，与指向字符串类似。

4.2.5 编辑和设置桩函数

当被测函数中存在函数调用时，需要为被测函数添加桩函数。默认地，在中  
AUnit 每生成一个用例时，对应每个被调用到的函数，都会为被测函数添加一个  
空的桩，方便用户在不进行任何桩操作的情况下，也能正常运行用例。

一般情况下，都需要用户手动去创建桩函数。这时，双击用例中**桩函数管理**对应的单元格（如 图 28 所示），打开**桩序列管理**对话框（如 图 29 所示）。

<div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div>				
	用例1		用例2	
get_2_3_f	输入	预期	输入	预期
用例描述				
全局变量				
negative(unsigned char)				
用户变量				
参数				
a(unsigned int)	2		2	
b(unsigned int)	2		2	
返回值				
returnValue(unsigned char)		1		1
桩函数管理	{{0}}		{{0}}	
ufabs				

图 28 桩函数管理-打开桩管理对话框

对话框中左侧区域（**桩函数库**）会显示出所有的桩，用户根据需要选择合适的桩添加到右侧区域（**桩序列**）中，点**确定**就完成了桩的添加，此时会在**用例设计编辑器**的桩函数管理下看到刚添加的桩函数。

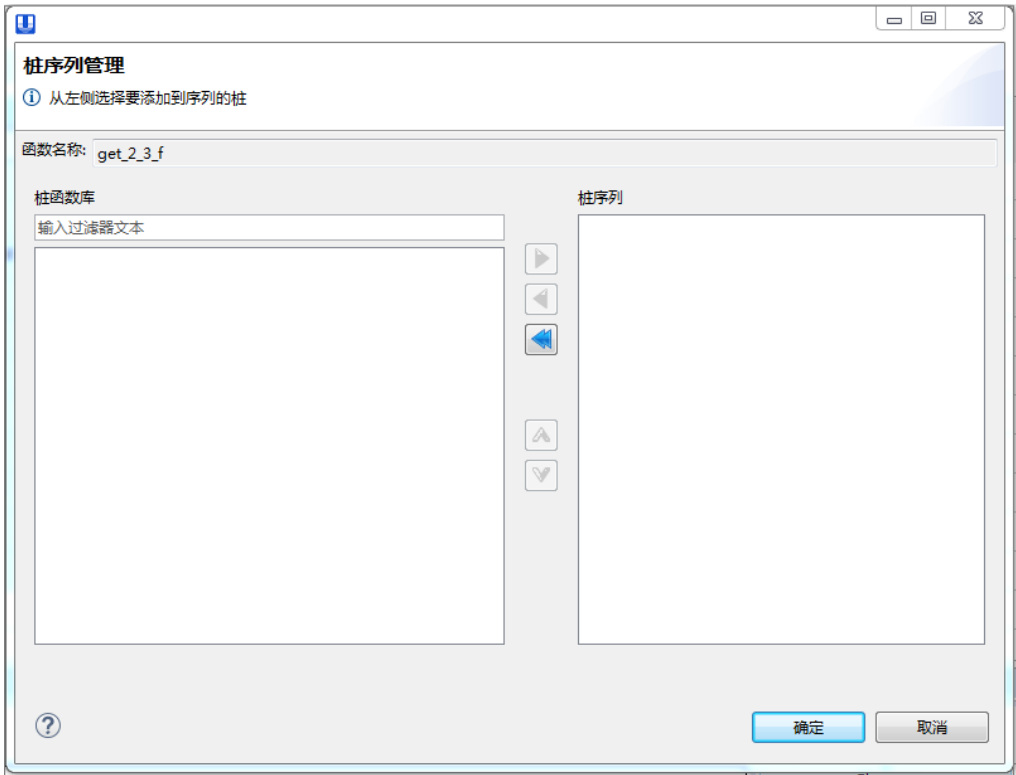


图 29 桩序列管理对话框

也可以选中**桩函数管理**，右击弹出快捷菜单，选中**添加桩函数**选项（如图 30 所示）。

	用例1		用例2	
get_2_3_f	输入	预期	输入	预期
用例描述				
全局变量				
negative(unsigned char)				
用户变量				
参数				
a(unsigned int)	2		2	
b(unsigned int)	2		2	
返回值				
returnValue(unsigned char)		1		1
桩函数管理	{0}		{0}	

编辑全局变量

添加桩函数

删除桩函数(不打桩)

图 30 右键菜单-添加桩函数

此时，如果还有被调函数没有打过桩，就会弹出**添加桩函数**对话框，只需在对话框中选择需要添加的桩函数并**确定**即可（如图 31 所示），然后在编辑器中**桩**

函数管理下就会看到刚添加的桩。否则，会弹出“已经为所有被调函数打过桩”的提示对话框。

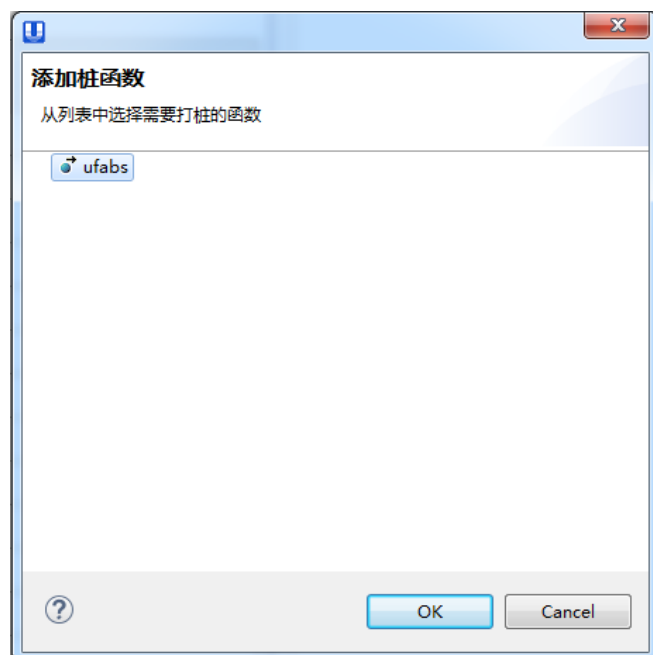


图 31 添加桩函数对话框

如果要删除桩，先在**桩函数管理**下选中要删除的桩，在右键菜单中选**删除桩函数**选项，就会将该桩函数从所有用例中移除，如图 32 所示。

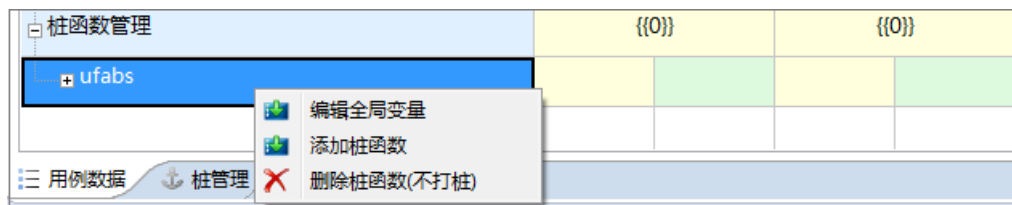


图 32 删除桩函数

## 4.2.6 设置输入值和输出预期

- 1) **设置输入值**：为输入变量赋值，如果是基本类型变量，直接手动输入值即可；如果是指针变量，可以用右键菜单来完成赋值（参看 4.2.4 小节），也可双击**输入单元格**，手动输入。如果所有输入都不赋值，运行时 AUnit 都会根据变量数据类型，给出默认值，保证用例可正常运行。
- 2) **设置输出预期**：如果需要对输出值（注意，包括桩的输出）进行预期，则在输入变量对应的**预期单元格**中输入预期值，如果不输入任何预期值，则不会进行预期比较。

示例如 图 33 所示。











<div></div>			用例1	
is_square	输入	预期		
用例描述				
全局变量				
用户变量				
参数				
rect(rectangle)				
rect.no(unsigned char)	'1'	'1'		
rect.length(unsigned int)	5	4		
rect.width(unsigned int)	4	4		
返回值				
returnValue(int)		0		
桩函数管理			{{0}}	

图 33 设置输入和预期

4.2.7 复制测试用例

AUnit 提供了用例复制功能。因为用例之间往往只有部分输入值是不同的，因此在手动设计用例时，只需选中一个已经编辑好的用例，点击按钮，即可轻松复制一个用例，然后再对需要修改的输入值进行编辑即可，可极大提高用例设计速度，如图 34 所示。

用例设计				
<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>				
	复制用例	用例1	用例2	
cal_girth	输入	预期	输入	预期
用例描述				
全局变量				
+ rects(rectangle [3])				
用户变量				
rect_ptr(rectangle)				
rect_ptr.no(unsigned char)	'1'		'1'	
rect_ptr.length(unsigned int)	10		10	
rect_ptr.width(unsigned int)	3		3	
参数				
+ rect(rectangle *)[NULL]	NULL		NULL	
返回值				
returnValue(int)		26		26
桩函数管理		{{0}}		{{0}}

图 34 复制用例

4.2.8 源码浏览视图辅助用例设计

手动设计用例时，一般需要参照源码来设计，因此 AUnit 提供了源码浏览视图来辅助用例设计，如图 35 所示。如果源码浏览视图未打开，可选择菜单栏中工具(T)->显示视图(V)->源码浏览打开视图。然后双击测试管理视图源程序中的某个待测函数，即可在源码浏览视图中定位到该函数。

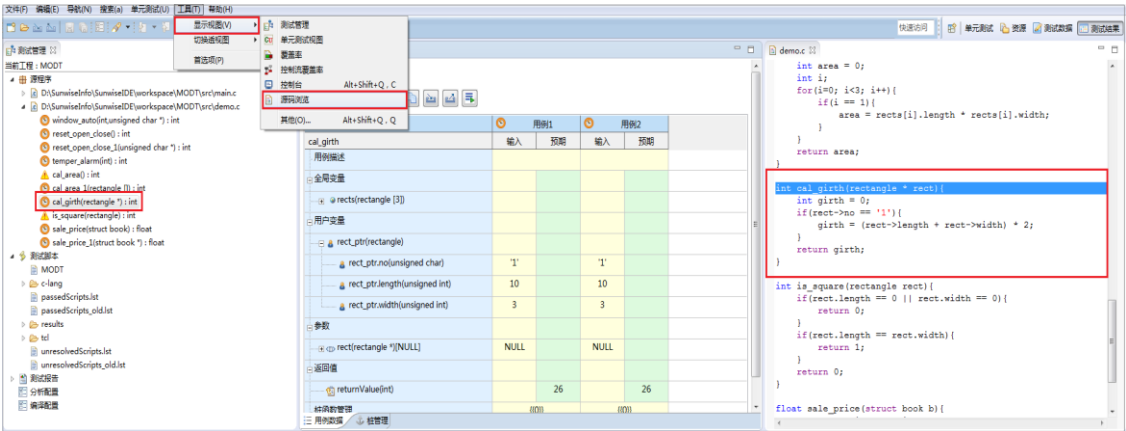


图 35 源码浏览辅助用例设计



## 5 运行单元测试及结果查看

设计好测试用例后便可运行测试并查看测试结果。

### 5.1 运行单元测试

设计好测试用例后，就可以构建并运行测试用例。在快速入门一章的产品使用一节（3.2.3）中已经提到过两种运行测试用例的途径。在选择任何一种方式运行测试用例之前，都需要先选中测试管理视图中的待测函数，然后点击工具栏的**重新构建工程**按钮，完成工程构建，如图 36 所示。

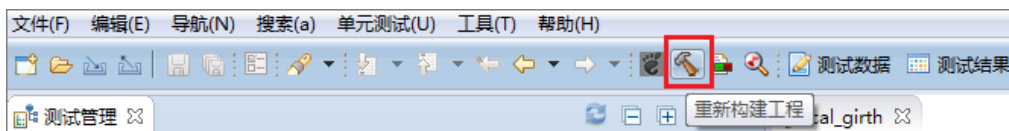


图 36 构建工程

两种方式的详细的操作如图 37 所示：

- **一键式运行测试用例：**快速一键运行测试用例，将生成用例脚本和执行脚本一步完成。在执行之前需要在高级配置中配置一下工程需要生成的脚本类型（如 Tcl、C 和 Python），便可方便快捷一键式运行测试。
- **分步骤运行测试用例：**分步骤运行测试用例，先在测试管理视图中选中待测函数，然后在右键菜单中选择**生成用例脚本**选项，根据要生成脚本的类型选择相应的子菜单，即生成脚本；生成脚本后，在右键菜单中选择**运行测试用例**选项，按照类型选择对应子菜单即开始运行测试用例。

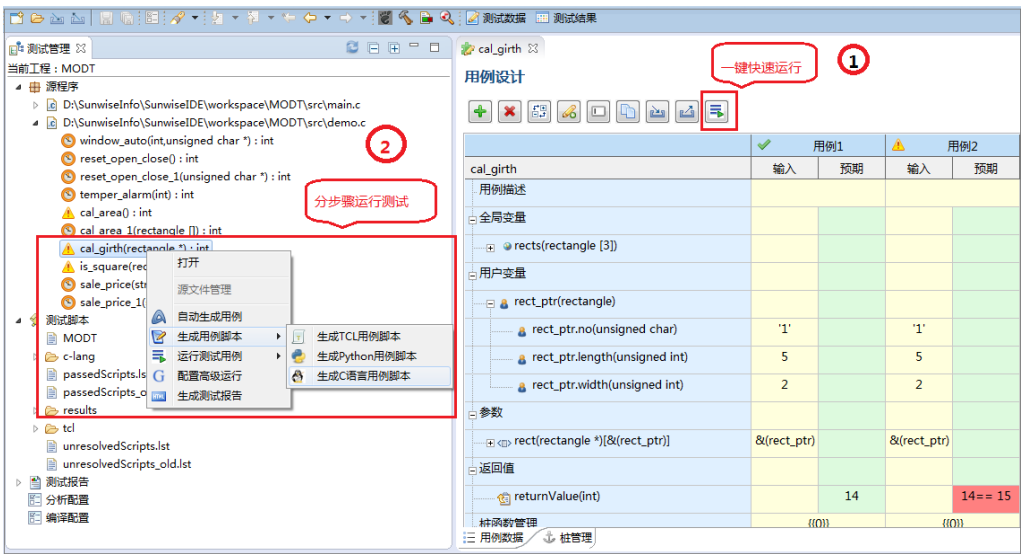


图 37 运行测试用例

## 5.2 查看测试结果

运行完测试用例后会生成测试结果。测试结果主要包括单元测试视图中的用例执行情况、覆盖率视图中的函数覆盖率信息、控制流覆盖视图中的图形化覆盖率展示、源码浏览视图中的代码覆盖展示、用例设计编辑器中的用例执行状态及预期失败状态展示以及测试管理视图中的函数测试执行状态等。

### 5.2.1 查看函数测试状态

测试运行完成后，函数的测试状态会在两个视图中体现出来，分别是单元测试视图和测试管理视图。在单元测试视图中，如果执行完后，进度条为绿色，说明所有用例执行成功，函数测试状态为成功，如图 38 所示；如果为红色，说明有用例执行错误或失败，则函数测试状态为失败，如图 39 所示。

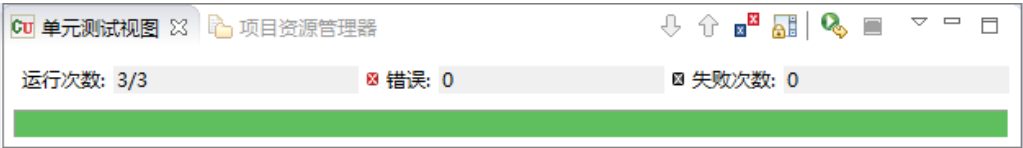


图 38 函数测试状态-成功

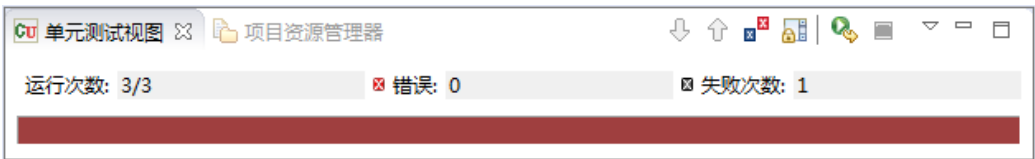






图 39 函数测试状态-失败

在测试管理视图中，如果待测函数名称前的图标为，表示尚未执行；如果图标为，表示测试成功；如果图标为，表示有用例执行失败但无错误，函数测试状态为失败；如果图标为，表示有用例执行错误，函数测试状态同样为失败，如图 40 所示。

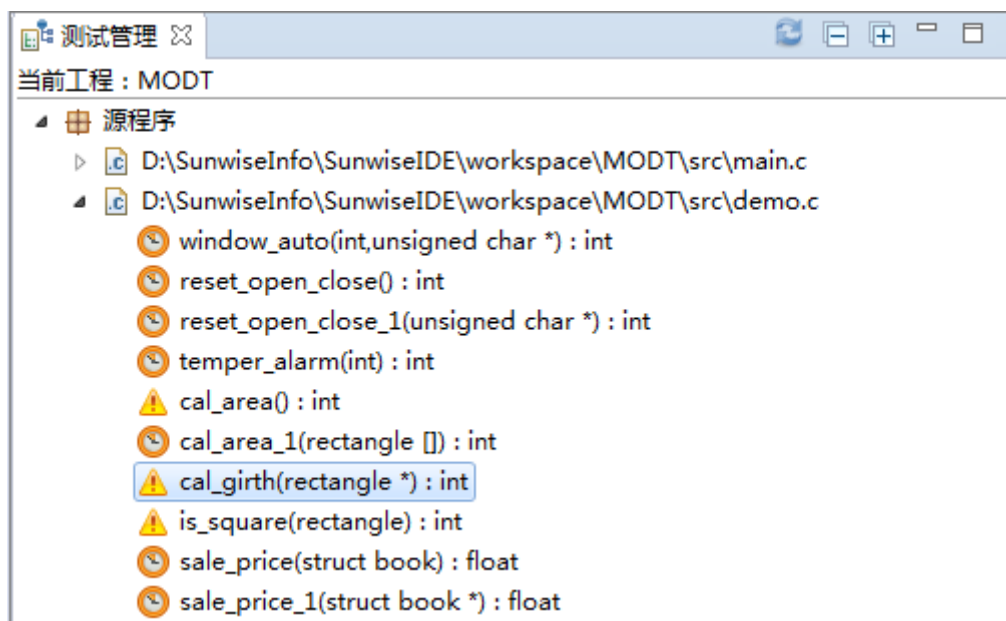


图 40 测试管理视图-测试状态

## 5.2.2 查看用例执行状态

运行完测试用例后，各个用例的执行状态会反映到两个地方，分别是**单元测试视图**和**用例设计编辑器**。在运行测试用例过程中，**单元测试视图**会实时显示用例总数、当前运行错误数、运行失败数、用例执行进度等信息。运行完成后，测试用例树会展开并定位到第一个运行失败的用例上，这里可以看到哪些用例运行成功，哪些运行失败。在**故障跟踪**中可查看失败或错误跟踪信息，如图 41 所示。

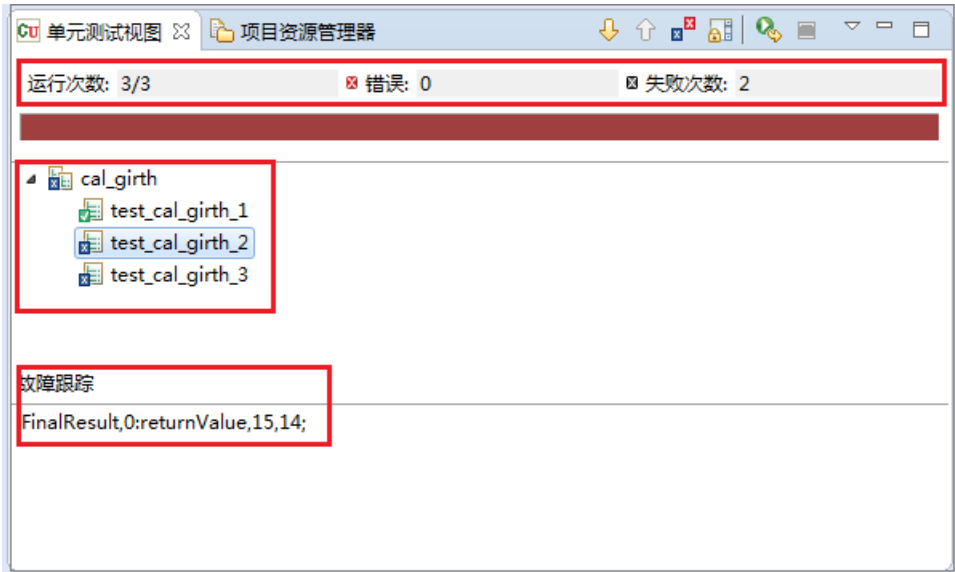






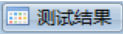
图 41 单元测试视图-用例状态

在用例设计编辑器中的用例名称前会以四种图标来表示用例的测试状态。其中表示尚未测试，表示测试成功，表示测试失败，表示运行错误，如图 42 所示。

	用例1	用例2	用例3			
cal_girth	输入	预期	输入	预期	输入	预期
用例描述						
全局变量						
rects(rectangle [3])						
用户变量						
rect_ptr(rectangle)						

图 42 用例编辑器-用例状态

### 5.2.3 查看函数覆盖率

函数覆盖率信息会在三个视图中体现出来，分别是覆盖率视图、控制流覆盖视图和源码浏览视图。点击工具栏中的按钮，将透视图切换到测试结果状态，就可看到上述三个视图。其中覆盖率视图中显示了被测函数及其所在源文件和工程的详细覆盖率信息，包括：语句覆盖率、语句覆盖数、语句未覆盖数、分支覆盖率、分支覆盖数和分支未覆盖数等，如图 43 所示。

[illegible]

图 43 覆盖率视图

控制流覆盖视图以结构化图形的方式展示函数覆盖情况，其中未覆盖的块以红色表示，覆盖到的块以绿色表示，这样就可以清楚看出哪些分支覆盖到，哪些未覆盖到，如图 44 所示。

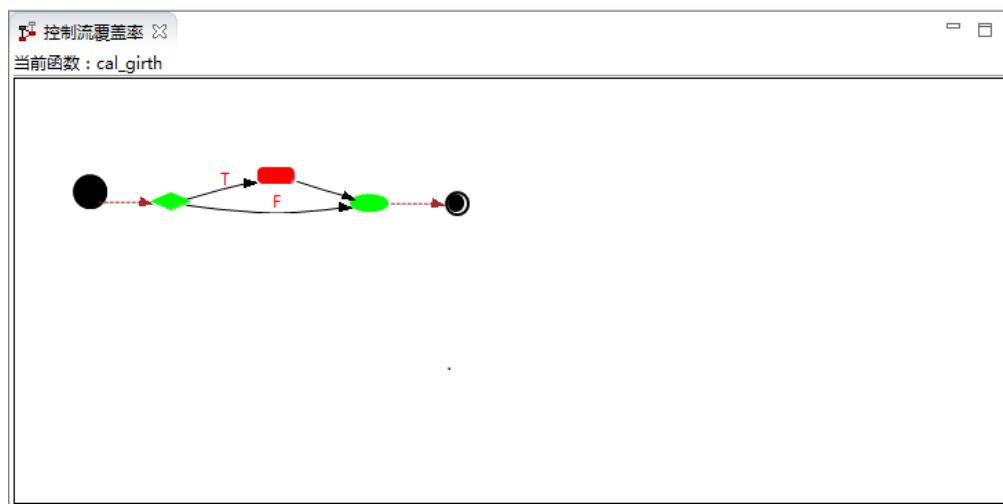


图 44 控制流覆盖视图

源码浏览视图中未覆盖的语句会以红色高亮方式标记出来，如图 45 所示。

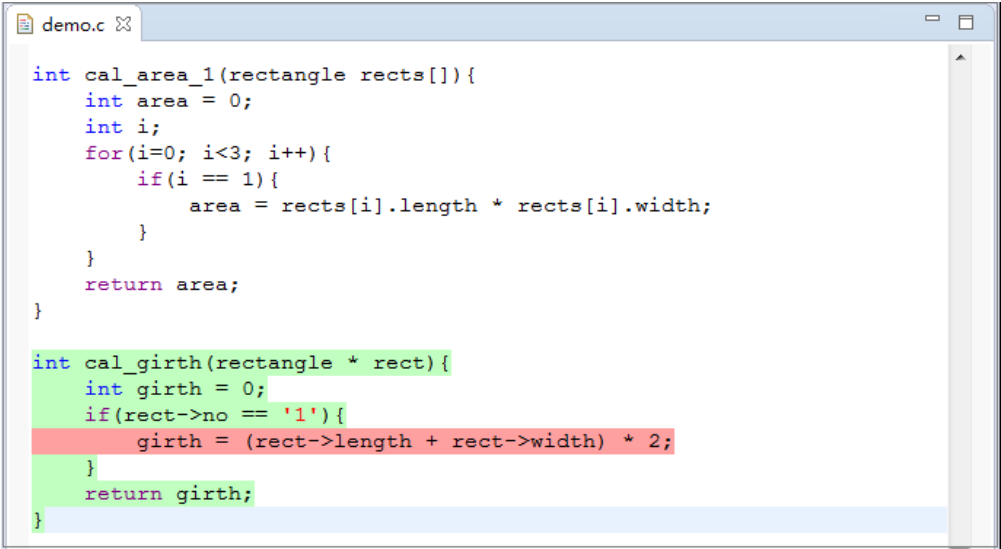


图 45 源码浏览视图覆盖展示

5.2.4 查看预期失败用例

对于运行失败的用例，在用例设计编辑器中，会将所有预期失败的输入在预期一列对应的单元格中以红色高亮方式显示，会同时显示出实际值和预期值。如图 46 所示。

	✓	用例1	⚠	用例2	⚠	用例3
cal_girth	输入	预期	输入	预期	输入	预期
用例描述						
全局变量						
rects(rectangle [3])						
用户变量						
rect_ptr(rectangle)						
rect_ptr.no(unsigned char)	'1'		'1'		'2'	
rect_ptr.length(unsigned int)	5		5		5	
rect_ptr.width(unsigned int)	2		2		2	
参数						
rect(rectangle *)[&(rect_ptr)]	&(rect_ptr)		&(rect_ptr)		&(rect_ptr)	
返回值						
returnValue(int)		14		14 == 15		0 == 14
桩函数管理		{(0)}		{(0)}		{(0)}

图 46 预期失败查看

5.3 生成并查看测试报告

AUnit 可以支持多种格式的可定制化测试报告导出，目前支持 HTML、Word 和 PDF。在**测试管理视图**中的任意区域右击弹出快捷菜单，选择**生成测试报告**选项，将一次生成所有函数的测试报告，如图 47 所示。

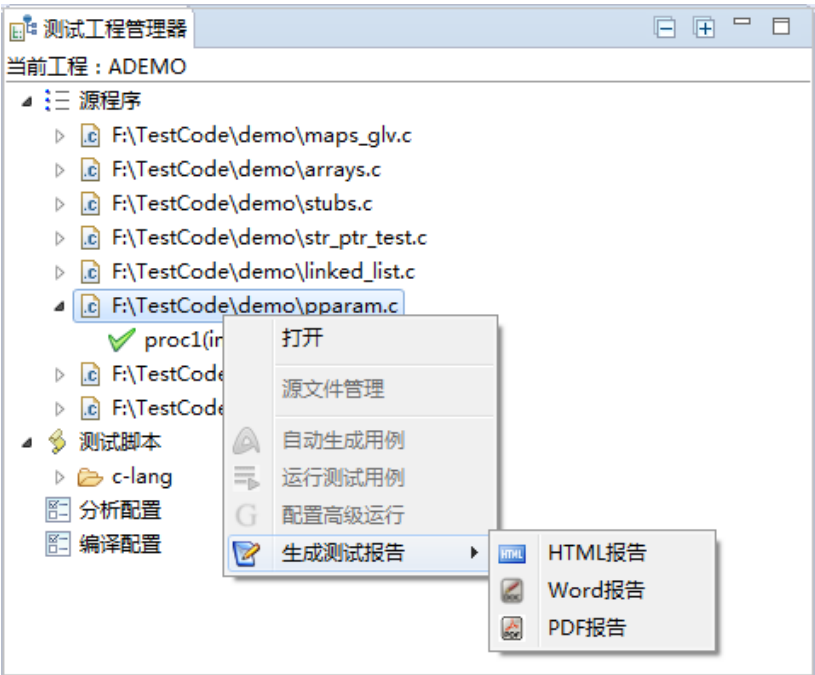


图 47 生成测试报告

报告生成后，会弹出提示对话框，提示报告是否成功生成，如果成功生成，会提示是否立即查看，以 HTML 格式报告为例，点击**确定**，会自动调用浏览器打开报告文件索引页，如图 48 所示。



图 48 HTML 报告-文件索引页

点击某个源文件链接，会跳转到该文件包含的函数索引页，如图 49 所示。

D:\SunwiseInfo\SunwiseIDE\workspace\MODT\src\demo.c					(返回首页)
序号	函数名	用例个数	已测试个数	语句覆盖率	
1	<a href="#">reset_open_close</a>	0	0	未统计	
2	<a href="#">sale_price</a>	0	0	未统计	
3	<a href="#">cal_area_1</a>	0	0	未统计	
4	<a href="#">cal_girth</a>	3	3	未统计	
5	<a href="#">is_square</a>	4	4	未统计	
6	<a href="#">reset_open_close_1</a>	0	0	未统计	
7	<a href="#">cal_area</a>	3	3	未统计	
8	<a href="#">window_auto</a>	1	0	未统计	
9	<a href="#">sale_price_1</a>	0	0	未统计	
10	<a href="#">temper_alarm</a>	0	0	未统计	

图 49 HTML 报告-函数索引页

点击任意函数链接，就会跳转到该函数对应的详细测试报告页，如图 50 所示。

单元测试报告

+

+

is\_square

(返回 demo.c 文件) (返回首页)

用例个数	运行用例数	失败用例数	语句覆盖率	分支覆盖率
4	4	2	未统计	

单元测试用例1

参数	参数名	输入	输出
	rect.width	3	
	rect.length	3	
	rect.no	'1'	
返回值类型	int	预期返回值	
测试结果	测试通过		

单元测试用例2

参数	参数名	输入	输出
	rect.width	3	
	rect.length	3	
	rect.no	'1'	
返回值类型	int	预期返回值	
测试结果	测试不通过 [returnValue: expect=1,actual=0]		

单元测试用例3

参数	参数名	输入	输出
	rect.width	4	
	rect.length	3	
	rect.no	'1'	
返回值类型	int	预期返回值	
测试结果	测试不通过 [returnValue: expect=0,actual=1]		

图 50 HTML 报告-函数详细报告页

也可以通过工具栏打开单元测试报告按钮来打开测试报告，如所示。



图 51 工具栏-打开单元测试报告



## 6 工程导入导出

SunwiseAUnit 支持将工作区中的测试工程导出到本地的归档文件或文件系统，同样也支持将本地归档文件或文件系统中的测试工程导入到工作区中。

### 6.1 导入工程

暂无。

### 6.2 导出工程

暂无。

## 7 产品更新

暂无。

## 8 AUnit 支持的环境

支持的运行平台：

- Microsoft Windows 9x/NT/2000/XP/7/8/10
- Linux
- Unix
- Mac OS X

支持的编译器：

- ARM
- Keil
- GNU GCC
- Green Hills
- WindRiver
- Visual Studio
- QNX

## 9 常见问题汇总

暂无。