

单元测试计划

ID : LY - UT

刘林青 1354361

杨安琪 1353005

目录

[1 引言](#)

[1.1 目的](#)

[1.2 测试策略](#)

[2 测试项目](#)

[2.1 数据库模块](#)

[2.1.1 被测方法参考表：](#)

[2.1.2 不被测方法表：](#)

[2.1.3 用例分析与设计](#)

[1\) openDataBase 测试分析与设计](#)

[2\) modifyDataBase 测试分析与设计](#)

[2.2 收入支出模块](#)

[2.2.1 被测方法参考表](#)

[2.2.2 不被测方法参考表](#)

[2.2.3 用例设计与分析](#)

[1\) setValue 测试分析与设计](#)

[2\) removeValue 测试分析与设计](#)

[3\) setType 测试分析与设计](#)

[4\) getSum 测试分析与设计](#)

[5\) getSumIncome 测试用例设计与分析](#)

[6\) getSumExpense 测试用例设计与分析](#)

[7\) getSumByType 测试用例设计与分析](#)

[2.3 资产负债模块](#)

[2.4 界面模块](#)

[2.4.1 被测方法参考表](#)

[2.4.2 不被测方法参考表](#)

1 引言

1.1 目的

本文档为“随手记”记账软件的单元测试活动提供范围、方法、资源和进度方面的指导。

本文档的主要读者是测试经理、测试人员和开发人员。

1.2 测试策略

以类为单元，采用自底向上的单元测试策略，通过设计相应的驱动和桩的方法来测试类中的方法，在选择类的被测方法时，根据类的功能进行判定。非空非注释代码行数 LOC>10 的方法进行单元测试，其他方法不进行单元测试。

决定单元测试次序的依据是《软件概要设计说明》中的软件层次架构图，按照被调用关系，从最底层的类开始测试，逐步向上，直至所有类测试完毕。主要采用白盒测试的测试方法进行用例设计，并以黑盒测试方法（边界值法等）进行补充。

2 测试项目

根据《软件详细设计说明书》中的内容，单元测试的测试项目如下：

2.1 数据库模块

设计类标识：LLC_001

2.1.1 被测方法参考表：

方法标识符	方法名	代码行（LOC）
LLC_001_FUN_001	openDataBase(void)	11
LLC_001_FUN_002	modifyDataBase(const string &querySQL)	22
LLC_001_FUN_003	runSQL(const string &	25

	querySQL,QueryResult & res)	
LLC_001_FUN_004	clearResult(QueryResult & res)	9

2.1.2 不被测方法表：

方法标识符	方法名	代码行 (LOC)
LLC_001_FUN_005	DataBase()	4
LLC_001_FUN_006	~DataBase()	4
LLC_001_FUN_007	closeDataBase(void)	4
LLC_001_FUN_008	rmDataBase(void)	4
LLC_001_FUN_009	createDataBase(void)	5

2.1.3 用例分析与设计

1) openDataBase 测试分析与设计

略

2) modifyDataBase 测试分析与设计

(1) 标识符定义 : UT_TD_001_002

(2) 被测特性 :

(3) 测试方法 :

a. 强健壮等价类法

采用等价类的测试方法，对插入收入项目这一功能进行测试，按照输入变量划分等价类。

输入变量有：（ID 为自动设置），日期，项目名称，金额，项目类型代码，项目备注。

将等价类划分如下：（蓝色为无效情况）

日期 D1:

D11={19800101~20991231 内的合法日期};

D12={19800101~20991231 内的不合法日期};

D13={小于 19800101 的日期};

D13={大于 20991231 的日期};

项目名称 D2:

D21={中文字符串};

D22={英文字符串};

D23={中英文混合字符串};

D24={空};

金额 D3:

D31={正整数, 用+标识};

D32={正数且两位小数, 用+标识};

D33={正数且两位以上小数, 用+标识};

D34={正整数, 不用+标识};

D35={正数且两位小数, 不用+标识};

D36={正数且两位以上小数, 不用+标识};

D37={负整数, 用-标识};

D38={负数且两位小数, 用-标识};

D39={负数且两位以上小数, 用-标识};

项目类型代码 D4:

D41={整数};

D41={小数};

项目备注 D5:

D51={中文字符串};

D52={英文字符串};

D53={中英文混合字符串};

D54={空};

b. 基路径测试法

采用基路径的测试方法, 对插入收入类型这一功能进行测试, 以源代码作为基础, 导出程序图, 如图 1-1-2 所示:

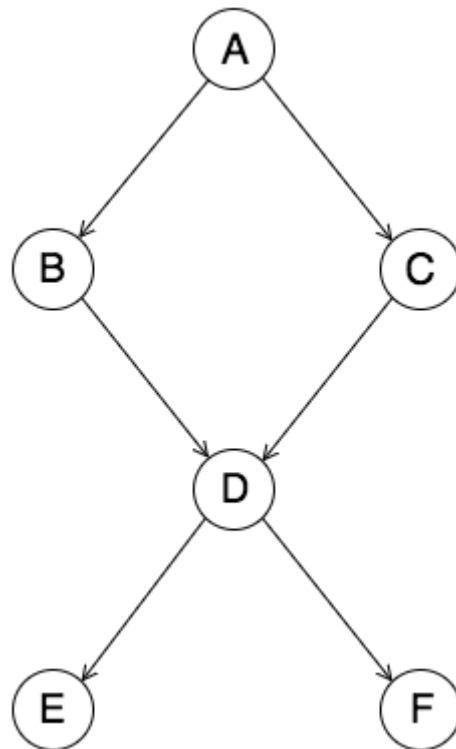


图 1-1-1 Database::modifyDatabase 程序图

依据 $V_G = e - n + 2 = 7 - 6 + 2 = 3$ ，基路径中应该包含 3 条独立的路径，选择一条作为基线路径并由此构成一组基路径，如下：

- 路径 1: A → B → D → E
- 路径 2: A → C → D → E
- 路径 3: A → B → D → F

满足上述三条基路径的测试用例如下，详细测试用例参见 UT_TD_001_002.xls

路径	输入		输出
	TypeID	TypeName	
1	不为零	不为空的字符串	插入成功
2	为零	不为空的字符串	插入失败
3	不为零	空	插入失败

(4) 测试用例：UT_TD_001_002.xls

(5) 测试通过标准：所有的用例都被执行，且测试结果与预期结果相同为通过，否则失败。

2.2 收入支出模块

设计类标识：LLC_002

2.2.1 被测方法参考表

方法标识符	方法名	代码行 (LOC)
LLC_002_FUN_001	bool setValue(Table & IncomeTable)	38
LLC_002_FUN_002	bool removeValue(const int & ID)	12
LLC_002_FUN_003	bool setType(Type & IncomeType)	21
LLC_002_FUN_004	float getSum(const int & beginTime, const int endTime);	9
LLC_002_FUN_005	float getSumIncome(const int &	10

	beginTime, const int endTime)	
LLC_002_FUN_006	float getSumExpense(const int & beginTime, const int endTime)	10
LLC_002_FUN_007	float getSumByType(const int & beginTime, const int endTime, const int & typeId);	13

2.2.2 不被测方法参考表

方法标识符	方法名	代码行 (LOC)
LLC_002_FUN_007	Income()	4
LLC_002_FUN_008	~Income()	4
LLC_002_FUN_009	string getErrorMessage()	4
LLC_002_FUN_010	float getSum()	4
LLC_002_FUN_011	int getTypeId(const string & typeName)	19
LLC_002_FUN_012	string getTypeName(const int & ID)	19
LLC_002_FUN_013	float Sum(const string & sql)	20
LLC_002_FUN_014	float getSumIncome()	4
LLC_002_FUN_015	float getSumExpense()	4
LLC_002_FUN_016	float getSumByType(const int & typeId)	7

2.2.3 用例设计与分析

1) setValue 测试分析与设计

(1) 标识符定义：UT_TD_002_001

(2) 被测特性：

- 输入日期不合法或不在指定范围内，数据库修改失败；
- 输入项目名称为空，数据库修改失败；
- 输入项目类型代码为小数时，数据库修改失败；
- 输入项目备注为空，数据库修改失败；
- 输入参数合法，数据库修改成功。

(3) 测试方法：

a. 基路径测试法:

以源代码作为基础，导出程序图，如图 2-1-1 所示:

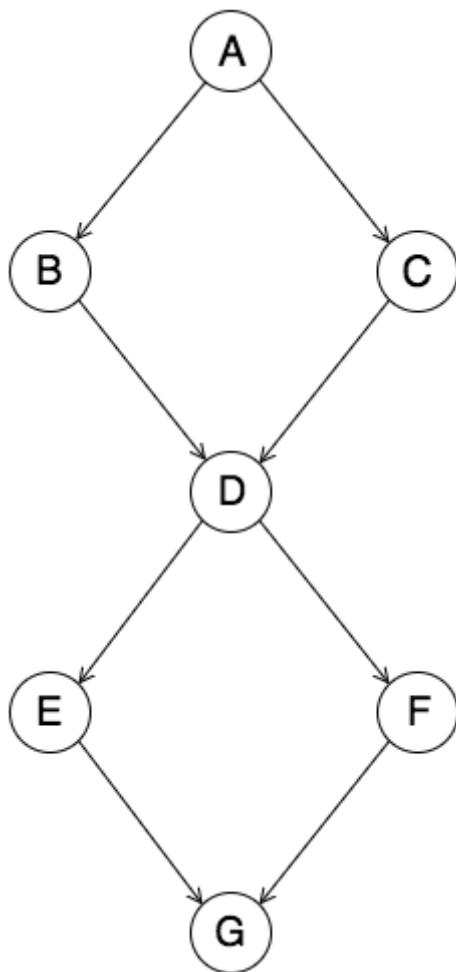


图 2-1-1 Income::setValue 程序图

依据 $V_G = e - n + 2 = 8 - 7 + 2 = 3$ ，基路径中应该包含 3 条独立的路径，选择一条作为基线路径并由此构成一组基路径，如下：

- 路径 1: A → B → D → E → G
- 路径 2: A → C → D → E → G
- 路径 3: A → B → D → F → G

满足上述三条基路径的测试用例如下，详细测试用例参见 UT_TD_002_001.xls

路径	输入		输出
	ID	remark 备注	
1	不为零	不为空的字符串	插入成功
2	为零	不为空的字符串	插入失败
3	不为零	空	插入失败

b. 强健壮等价类法：

采用强健壮等价类的方法进行测试，根据输入变量划分等价类。

输入变量有：（ID 为自动设置），日期，项目名称，金额，项目类型代码，项目备注。

将等价类划分如下：（蓝色为无效情况）

日期 D1：

D11={19800101~20991231 内的合法日期}；

D12={19800101~20991231 内的不合法日期}；

D13={小于 19800101 的日期}；

D13={大于 20991231 的日期}；

项目名称 D2：

D21={中文字符串}；

D22={英文字符串}；

D23={中英文混合字符串}；

D24={空}；

金额 D3：

D31={正整数，用+标识}；

D32={正数且两位小数，用+标识}；

D33={正数且两位以上小数，用+标识}；

D34={正整数，不用+标识}；

D35={正数且两位小数，不用+标识}；

D36={正数且两位以上小数，不用+标识}；

D37={负整数，用-标识}；

D38={负数且两位小数，用-标识}；

D39={负数且两位以上小数，用-标识}；

项目类型代码 D4：

D41={整数}；

D41={小数}；

项目备注 D5：

D51={中文字符串}；

D52={英文字符串}；

D53={中英文混合字符串}；

D54={空}；

（4）测试用例：参见 UT_TD_002_001.xls

（5）测试通过 / 失败标准：

所有的用例都被执行，且测试结果与预期结果相同为通过，否则失败。

2) removeValue 测试分析与设计

（1）标识符定义：UT_TD_002_002

（2）被测特性：

- 要删除的 ID 超过已存在记录范围，删除失败；

- 要删除的 ID 为小数，删除失败；
- 要删除的 ID 为负数，删除失败；
- 要删除的 ID 合法，删除成功。

(3) 测试方法：

a. 条件覆盖法

以源代码作为基础，分析每个条件可能的取值：

- ID = 0，取真值，记为 T；
- ID <> 0，取假值，记为 F。

给出两个测试用例，详细测试用例参见 UT_TD_002_002.xls

路径	输入 ID	输出
1	合法	删除成功
2	不合法	删除失败

b. 边界值法

采用边界值的测试方法，执行 UT_TD_002_001.xls 之后，Income 表中共有 81 条收入支出记录，因此 ID 为 [1, 81] 范围内的整数，

由此取得 5 个有效值：

$D_{min}=1$, $D_{minn+}=3$, $D_{nom}=49$, $D_{max-}=77$, $D_{max}=81$;

无效值为：

$D=\{0, 100, -1, 1.34\}$

(4) 测试用例：参见 UT_TD_002_002.xls

(5) 测试通过 / 失败标准：

所有的用例都被执行，且测试结果与预期结果相同为通过，否则失败。

3) setType 测试分析与设计

(1) 标识符定义：UT_TD_002_003

(2) 被测特性：

- 输入项目类型 ID 为负数时，新建项目类型失败；
- 输入项目类型 ID 为小数时，新建项目类型失败；
- 输入项目类型名称为空时，新建项目类型失败；
- 输入参数合法时，新建项目类型成功。

(3) 测试方法：

a. 基路径测试法

以源代码作为基础，导出程序图，如图 2-1-2 所示：

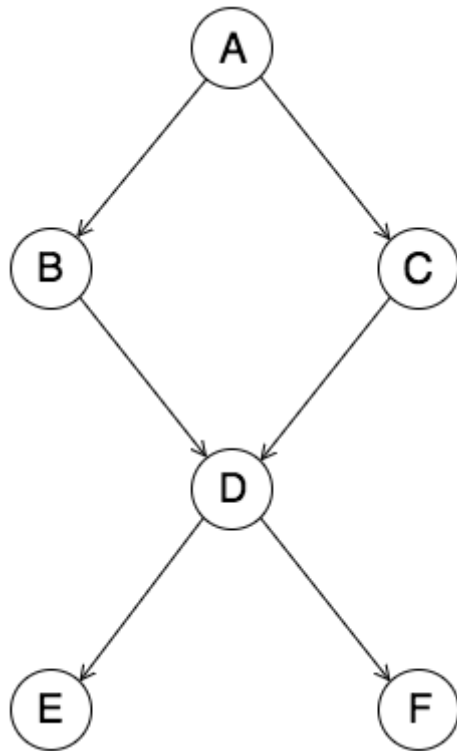


图 2-1-1 Income::setValue 程序图

依据 $V_G = e - n + 2 = 7 - 6 + 2 = 3$ ，基路径中应该包含 3 条独立的路径，选择一条作为基线路径并由此构成一组基路径，如下：

- 路径 1: A → B → D → E
- 路径 2: A → C → D → E
- 路径 3: A → B → D → F

满足上述三条基路径的测试用例如下，详细测试用例参见 UT_TD_002_003.xls

路径	输入		输出
	TypeID	TypeName	
1	不为零	不为空的字符串	插入成功
2	为零	不为空的字符串	插入失败
3	不为零	空	插入失败

b. 强健壮等价类法：

采用强健壮等价类的方法进行测试，根据输入变量划分等价类。

输入变量有：项目类型 ID，项目类型名称

将等价类划分如下：（蓝色为无效情况）

项目类型代码 D1：

D11={整数};

D12={小数};

D13={负数};

项目名称 D2:

D21={中文字符串};

D22={英文字符串};

D23={中英文混合字符串};

D24={空};

(4) 测试用例：参见 UT_TD_002_003.xls

(5) 测试通过 / 失败标准：

所有的用例都被执行，且测试结果与预期结果相同为通过，否则失败。

4) getSum 测试分析与设计

(1) 标识符定义：UT_TD_002_004

(2) 被测特性：

- 开始时间不合法，查询失败；
- 开始时间不在 1980/01/01~2099/12/31 范围内，查询失败；
- 结束时间不合法，查询失败；
- 结束时间不在 1980/01/01~2099/12/31 范围内，查询失败；
- 开始时间晚于结束时间，查询失败；
- 输入参数合法，查询成功。

(3) 测试方法：

采用等价类的测试方法，按照输入变量划分等价类。

输入变量有：开始时间 begin，结束时间 end。划分等价类如下：（蓝色为无效情况）

D1= { begin<=end } ；

D2= { begin>end } ；

D1 这个等价类包含合法查询与不合法查询两种情况，进一步划分等价类如下：

D11= { begin<=end , begin 和 end 均为 1980/01/01 ~ 2099/12/31 内合法日期 }

D12= { begin<=end , begin 为不合法日期 , end 符合要求 }

D13= { begin<=end , begin 为小于 1980/01/01 的日期 , end 符合要求 }

D14= { begin<=end , begin 为大于 2099/12/31 的日期 , end 符合要求 }

D15= { begin<=end , end 为不合法日期 , begin 符合要求 }

D16= { begin<=end , end 为小于 1980/01/01 的日期 , begin 符合要求 }

D17= { begin<=end , end 为大于 2099/12/31 的日期 , begin 符合要求 }

(4) 测试用例：参见 UT_TD_002_004.xls

(5) 测试通过 / 失败标准：

所有测试用例都执行，且测试结果与预期相同为通过，否则失败。

5) getSumIncome 测试用例设计与分析

测试用例的设计思路与 7) 相同，此处略去。测试用例参见 UT_TD_002_005.xls 。

6) getSumExpense 测试用例设计与分析

测试用例的设计思路与 7) 相同，此处略去。测试用例参见 UT_TD_002_006.xls 。

7) getSumByType 测试用例设计与分析

测试用例的设计思路与 7) 相同，此处略去。测试用例参见 UT_TD_002_007.xls 。

2.3 资产负债模块

与收入支出模块测试用例设计思路相同，略。

2.4 界面模块

2.4.1 被测方法参考表

2.4.2 不被测方法参考表

方法标识符	方法名	代码行 (LOC)
LLC_005_FUN_001	void initDataBase(void)	21
LLC_005_FUN_002	void backupDataBase(void)	23
LLC_005_FUN_003	bool judgeTime(int time)	63
LLC_005_FUN_004	bool inputIncomeData(Table & data)	63
LLC_005_FUN_005	void addIncome(void)	15
LLC_005_FUN_006	void removeIncome(void)	20
LLC_005_FUN_007	void outputResult(const string & sql, const int which)	72
LLC_005_FUN_008	void searchIncomeByName(void)	12
LLC_005_FUN_009	void searchIncomeByTime(void)	30

LLC_005_FUN_010	bool inputInvestmentData(Table & data)	63
LLC_005_FUN_011	void addInvestment(void)	15
LLC_005_FUN_012	void removeInvestment(void)	20
LLC_005_FUN_013	void searchInvestmentByName(void)	12
LLC_005_FUN_014	void searchInvestmentByTime(void)	31
LLC_005_FUN_015	void changeSocietyInsurance()	83
LLC_005_FUN_016	void outputSocietyInsuranceState()	97
LLC_005_FUN_017	void outputCashFlow(void)	79
LLC_005_FUN_018	void outputBalanceSheet(void)	79
LLC_005_FUN_019	void analysisState(void)	102
LLC_005_FUN_020	void outputDatabaseState(void)	38