

《软件工程》

2023年春

第七讲 软件实现与测试

内容

一．软件实现基础

二．编写代码

三．软件测试



二、编写代码

内容

1. 编写代码

- ✓任务、过程和方法
- ✓代码片段的重用

2. 软件缺陷和调试

- ✓软件缺陷、错误和失效
- ✓代码缺陷的应对方法及调试

3. 解决编程和调试问题

- ✓开源技术问答社区
- ✓群智知识的利用



1.1 编写代码的任务

□根据软件设计信息，借助于程序设计语言，编写出目标软件系统的**源程序代码**，开展**程序单元测试**、**代码审查**等质量保证工作

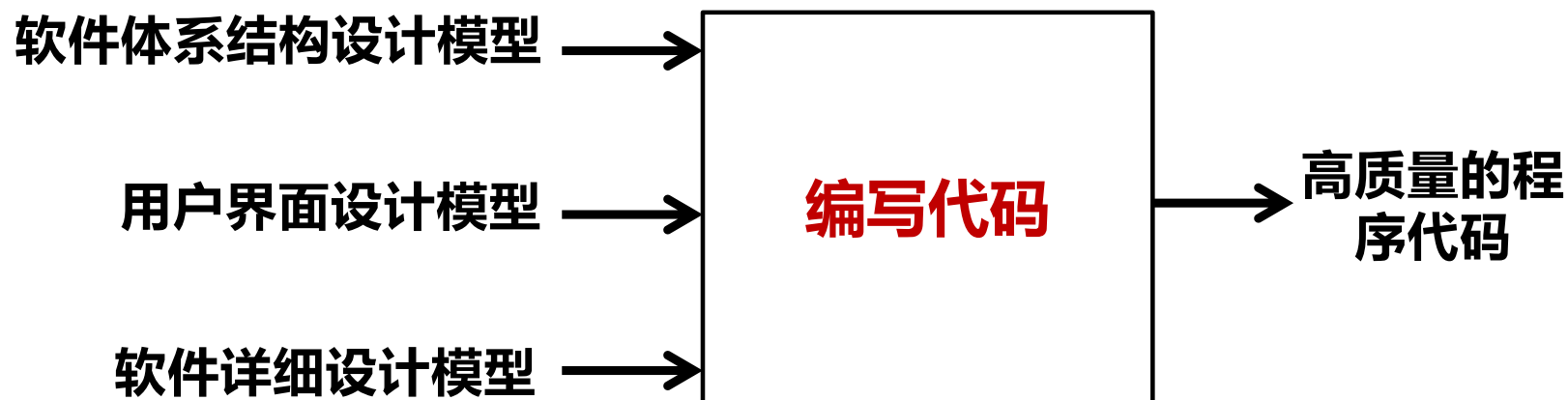
✓编写代码既是一个生成代码的过程，也是对生成的代码进行质量保证的过程

□兼具**软件创作**和**软件生产**的过程

✓自由地开展代码创作，编写出满足要求的程序代码，发挥其创新性和主观能动性，创作出算法精巧、运行高效的代码

✓按照软件质量保证的规范和要求，生产出高质量的代码。程序员需要约束其编程行为，防止随意性、自由性的编程活动，确保其编程活动及其所产生的程序代码满足工程化开发的要求

编写代码的任务



1.1.1 编写类代码

- ① 编写实现类的代码
- ② 编写实现类方法的代码
- ③ 编写实现类间关联的代码
- ④ 编写实现设计类间聚合和组合关系的代码
- ⑤ 编写实现接口关系的代码
- ⑥ 编写实现继承关系的程序代码
- ⑦ 编写实现包的代码

(1) 编写实现类的代码

□ **设计模型**（如设计类图）详细描述了软件系统中类的详细设计信息，包括可见性、类名、属性、方法等等

□ 程序员需要将这些设计信息直接转换为用程序设计语言表示的**实现结构和代码**

```
public class User {  
    private String account; //用户的账号  
    private String password; //用户的密码  
    private String name; //用户的名字  
    private String mobile; //用户的移动手机号  
    private int type; //用户的类别  
  
    public void User(String account, String password); //构造函数  
    public void User(String account, String password,   
        String name, String mobile, int type);  
    public String getUserName( ); //获取用户的名字  
    public String getUserAccount( ); //获取用户的账号  
    public int getUserType( ); //获取用户的类别  
    public String getUserMobile( ); //获取用户的手机号  
    public void setUserPsw(String userPsw); //设置用户的密码  
}
```

(2) 编写实现类方法的代码

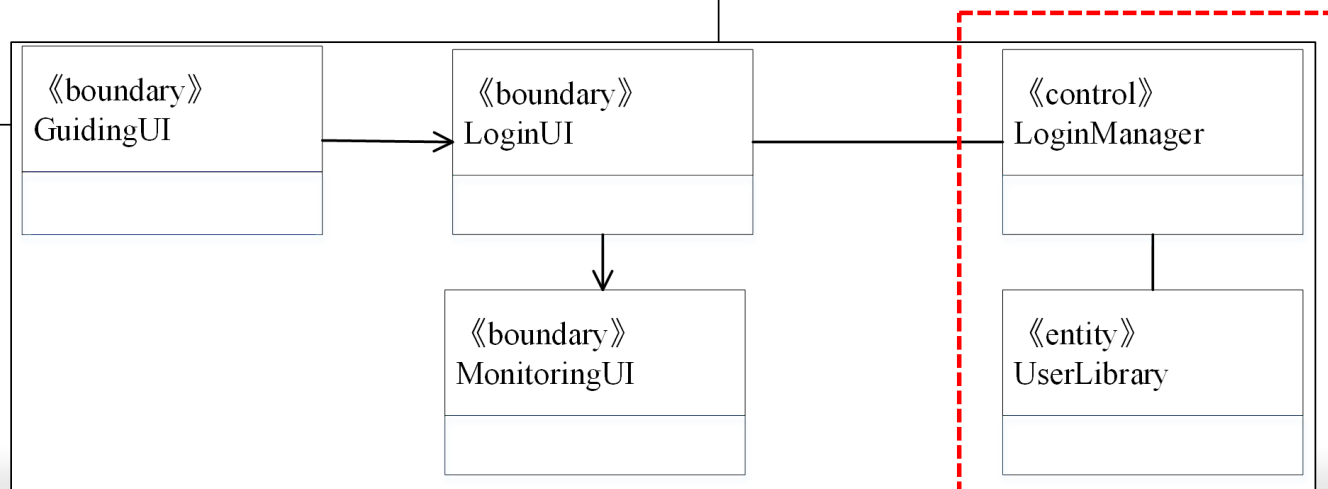
□ 基于类方法的设计描述 (UML的活动图表示), 程序员可以依此为依据来编写类方法的实现代码

```
public int login(String account, String password) {  
    final int ERROR_ACCOUNT_EMPTY = 1; //表示账号为空的错误代码  
    final int ERROR_PASSWORD_EMPTY = 2; //表示密码为空的错误代码  
    final int ERROR_INVALID_USER = 3; //表示用户非法的错误代码  
    final int LOGIN_SUCCESS = 0; //表示用户合法的代码  
    int result;   
      
    If (account.getLength() == 0) { //检查 account 是否为空串  
        result = ERROR_ACCOUNT_EMPTY; //表示账号为空  
    } else if (password.getLength() == 0) { //检查 password 是否为空串  
        result = ERROR_PASSWORD_EMPTY; //表示密码为空  
    } else {  
        //向 UserLibrary 对象发消息以验证用户的身份是否合法  
        boolean validUser = userLib.isUserValid(account, password);  
        If (validUser) {  
            result = LOGIN_SUCCESS;  
        } else {  
            result = ERROR_INVALID_USER;  
        }  
    }  
    return result;  
}
```

(3) 编写实现类间关联的代码

□ 将类间关联关系的**语义信息**具体落实到相应类的**程序代码中**，即综合考虑关联关系的**方向性、多重性、角色名和约束特性**等信息来编写相关的类程序代码

```
public class LoginManager {  
    private UserLibrary userLib; // UserLibrary 的对象  
    ...  
}
```



(4) 编写实现设计类间聚合和组合关系的代码

- 可以采用类似于实现**关联关系的方法**来编写实现聚合和组合关系的代码
- 根据多重性来设计相应类属性的数据结构

(5) 编写实现接口关系的代码

- 类设计模型可能包含有表征类与接口之间实现关系的语义信息**
- 诸多面向对象程序设计语言（如Java、C++等）提供了专门针对接口实现的语言机制，因而可以直接将接口设计信息转换为相应的程序代码**
 - ✓如 “Implement” 机制

(6) 编写实现继承关系的程序代码

□面向对象程序设计语言（如Java、C++）提供了继承机制以及相应的语言设施

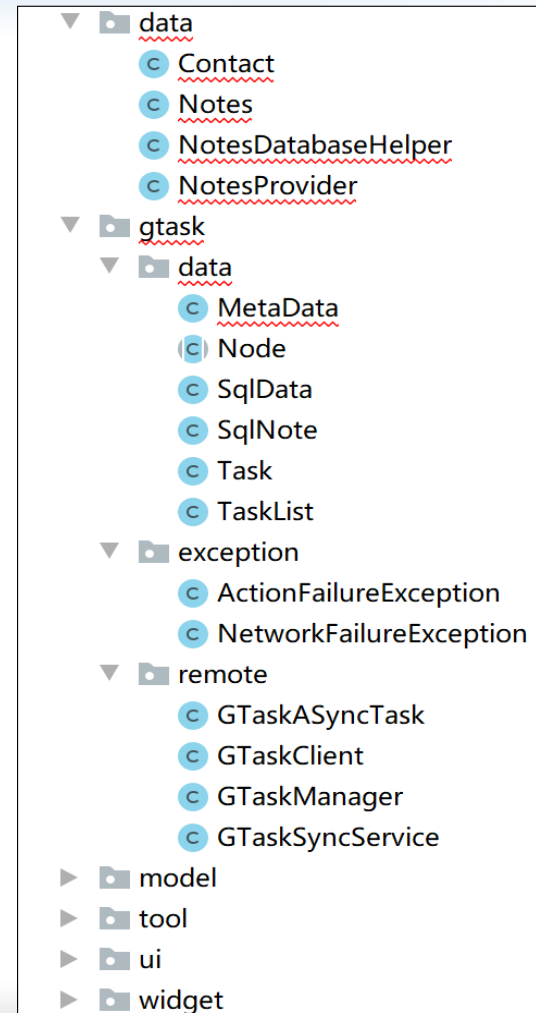
- ✓Java支持单重继承，C++支持多重继承
- ✓如“extends”机制

□将设计模型中的类间继承关系用程序设计语言提供的语言机制来表示

```
public class LoginUI extends Activity {  
    //成员方法说明  
    public void login();  
    public void cancel();  
    public boolean isInputAccountValid();  
    public boolean isInputPswValid();  
}
```

(7) 编写实现包的代码

- 用包 (package) 来**组织和管理**软件系统中的类
- 包是对软件系统中模块的**逻辑划分**，也可以将包视为是一种**子系统**
- 面向对象程序设计语言 (如Java) 提供了对包进行编程的语言机制，每个包对应于代码目录结构中的**某个目录**



1.1.2 编写用户界面代码

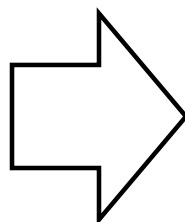
□用户界面设计模型

- ✓描述了构成用户界面的各个界面设计元素（包括静态元素、动态元素、用户输入元素、用户命令元素等）
- ✓用户界面之间的跳转关系

□编码实现

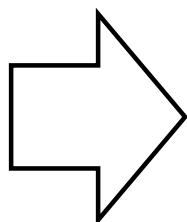
- ✓编写界面类属性的代码以定义界面设计元素
- ✓编写界面类的方法以对界面操作或者对界面事件进行响应处理

示例：“LoginUI” 的编码实现



```
public class LoginUI extends Activity {  
    private EditText mAccount;  
    private EditText mPsw;  
    private Button mCancelButton;  
    private Button mLoginButton;  
    private UserLibrary mUserLibrary;  
  
    public void onCreate(Bundle savedInstanceState) {  
        ...  
    }  
    //用户登录方法  
    public void login() {  
        ...  
    }  
    //取消登录方法  
    public void cancel() {  
        ...  
    }  
}
```

示例：“LoginUI” 的编码实现



```
OnClickListener mListener = new OnClickListener() {  
    public void onClick(View v) {  
        switch (v.getId()) {  
            case R.id.login_btn_cancel: //取消  
                onPause();  
                break;  
            case R.id.login_btn_login: //确认  
                login();  
                break;  
        }  
    }  
};
```

1.1.3 编写数据设计代码

□数据设计

- ✓定义了软件系统中需要持久保存数据及其组织（如数据库的表、字段）和存储（如数据库中的记录）方式
- ✓设计了相应的类及其方法来读取、保存、更新和查询持久数据

□编码实现

- ✓创建相应的数据库关系表格及其内部的各个字段选项等，确保它们满足设计的要求和约束
- ✓编写相应的程序代码来操作数据库，如增加、删除、更改、查询数据记录等

示例：“T_User”表的创建

```
private static final String TABLE_NAME = "T_User";  
public static final String USER_ACCOUNT = "user_account";  
public static final String USER_NAME = "user_name";  
public static final String USER_PSW = "user_password";  
public static final String USER_MOBILE = "user_mobile";  
public static final int USER_TYPE = "user_type";  
  
//创建数据库表“T_User”的 SQL 语句  
String DB_CREATE = "CREATE TABLE " + TABLE_NAME + " (" +  
    + USER_ACCOUNT + " varchar primary key," + USER_NAME + " varchar," +  
    + USER_PSW + " varchar," + USER_MOBILE + " varchar," +  
    + USER_TYPE + " integer" + ");";  
db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME + ";"); //执行 SQL 语句  
db.execSQL(DB_CREATE);
```

示例：连接和关闭数据库的代码

```
// 打开数据库↵
public void openDataBase() throws SQLException {↵
    mDatabaseHelper = new DataBaseManagementHelper(mContext);↵
    mSQLiteDatabase = mDatabaseHelper.getWritableDatabase();↵
}↵
// 关闭数据库↵
public void closeDataBase() throws SQLException {↵
    mDatabaseHelper.close();↵
}↵
```

示例：操作数据库的程序代码

```
//向数据库表中插入用户的数据↵  
public boolean insertUser(User user) {↵  
    String UserAccount = user.getUserAccount();↵  
    String UserName=user.getUserName();↵  
    String UserPsw=user.getUserPsw();↵  
    String UserMobile=user.getUserMobile();↵  
    int UserType=user.getUserType();↵  
    ContentValues values = new ContentValues();↵  
    values.put(USER_NAME, UserName);↵  
    values.put(USER_PSW, UserPsw);↵  
    values.put(USER_MOBILE, UserMobile);↵  
    values.put(USER_TYPE, UserType);↵  
    return mSQLiteDatabase.insert(TABLE_NAME, ID + "=" + UserAccount, values);  
}↵
```

示例：判断用户账号和密码合法性的代码

```
// 基于账号和密码来判断用户身份的合法性↵  
public boolean verifyUserValidity(String account,String psw){↵  
    boolean result= FALSE;↵  
    Cursor mCursor=mSQLiteDatabase.query(TABLE_NAME, null,  
USER_ID+"="+account+" and "+USER_PSW+"="+psw, null, null, ↵  
null, null);↵  
    if(mCursor!=null){↵  
        result=TRUE;↵  
        mCursor.close();↵  
    }↵  
    return result;↵  
}↵
```


1.2 代码片段的重用

□何为代码片段

- ✓对应于类代码中所包含的一组语句序列
- ✓实现了类中的一个具体、细粒度的功能

□代码片段示例

- ✓与远端数据库服务器建立连接
- ✓向远端的Socket程序发送一段数据

```
mDatabaseHelper = new DataBaseManagementHelper(mContext);  
mSQLiteDatabase = mDatabaseHelper.getWritableDatabase();
```


开源社区中的代码片段

- 开源技术问答社区（如Stack Overflow、CSDN）中，大量的程序员在其中分享了许多形式多样、极有价值的代码片段
- 通常这些代码片段都经过实践检验，因而表现出较高的代码质量
- 在编写代码的工程中，程序员可以针对其代码编写要求，到开源技术问答社区中寻找相关的代码片段，然后通过
对代码片段的理解，选定和重用所需的代码片段，进而完成相应的编程任务

示例：重用开源社区中的代码片段

□完成与MySQL数据库服务器连接的代码片段

```
12         try{
13             Class.forName("com.mysql.jdbc.Driver");
14         } catch (ClassNotFoundException e){
15             System.out.println("未能成功加载驱动程序, 请检查是否导入驱动程序!");
16             e.printStackTrace();
17         }
18         Connection conn = null;
19         try{
20             conn = DriverManager.getConnection(URL, NAME, PASSWORD);
21             System.out.println("获取数据库链接成功");
22         } catch (SQLException e){
23             System.out.println("获取数据库连接失败");
24             e.printStackTrace();
25         }
```

内容

1. 编写代码

- ✓任务、过程和方法
- ✓代码片段的重用

2. 软件缺陷和调试

- ✓软件缺陷、错误和失效
- ✓代码缺陷的应对方法及调试

3. 解决编程和调试问题

- ✓开源技术问答社区
- ✓群智知识的利用



2.1 何为软件缺陷

□软件缺陷是指软件制品中存在**不正确的软件描述和实现**

- ✓存在缺陷的软件制品不仅包括**程序代码**，而且还包括需求和设计的**模型和文档**
- ✓软件缺陷**产生于软件开发全过程**，只有有人介入的地方就有可能产生软件缺陷
- ✓**任何人**都有可能在软件开发过程中犯错误，进而引入软件缺陷
- ✓无论是高层的需求分析和软件架构缺陷还是底层的详细设计缺陷，它们**最终都会反映在程序代码**之中，导致程序代码存在缺陷

软件缺陷带来的问题：错误

□存在缺陷的程序代码在运行过程中会产生不正确或者不期望的**运行状态**，将这种情况称程序出现了错误

- ✓经过计算后某个变量的取值不正确
- ✓接收到的消息内容不正确
- ✓打开一个非法的文件

□引发程序报错



错误带来的问题：失效

- 运行错误的程序无法为用户提供**所需的功能和行为**，在此情况下我们称程序出现了**失效**
 - ✓如用户无法正常登录到系统中
 - ✓无法正确地分析出老人是否处于摔倒的状态等等。
- 程序**错误的根源**在于程序中存在**缺陷**，程序的错误运行必然导致软件失效
- 错误和失效是程序缺陷在程序运行时的**内部展示和外在工作表现**

2.2 软件缺陷的描述（1/2）

□标识符

- ✓每个软件缺陷都被给予一个唯一的标识符。

□类型

- ✓说明软件缺陷的类型，如**需求缺陷、设计缺陷、代码缺陷**
- ✓代码缺陷还可以进一步区分为是逻辑缺陷、计算缺陷、判断缺陷

□严重程度

- ✓**危急程度**是指缺陷会影响软件的正常运行甚至危及用户安全
- ✓**严重程度**的缺陷是指会导致软件丧失某些重要功能，或出现错误
- ✓**一般程度**的缺陷是指缺陷会使得软件丧失某些次要的功能
- ✓**轻微程度**是指缺陷会导致软件出现小毛病，但不影响正常运行

软件缺陷的描述（2/2）

- 症状**：软件缺陷所引发的程序错误是什么，有何运行表现
- 修复优先级**：缺陷应该被修复的优先程度，包括：非常紧迫、紧迫、一般和不紧迫等几种
- 状态**：描述缺陷处理的进展状态，如已经安排人员来处理、正在修复、修复已经完成等。
- 发现者**：谁发现了软件缺陷。
- 发现时机**：什么状况下发现的软件缺陷
- 源头**：软件缺陷源头在哪里，如软件文档的哪一个部分
- 原因**：说明导致软件缺陷的原因是什么

2.3 软件缺陷的应对方法（1/2）

□预防缺陷

- ✓通过运用各种软件工程技术、方法和管理手段，在软件开发过程中**预防和避免软件缺陷，减少软件缺陷的数量，降低软件缺陷的严重程度**
- ✓采用结对编程、严格的过程管理、必要的技术培训、CASE工具的使用等手段，起到预防缺陷的目的

□容忍缺陷

- ✓增强软件的缺陷容忍度，借助于**软件容错机制和技术**，允许软件出现错误，但是在**出现错误时软件仍然能够正常的运行**
- ✓在高可靠软件系统的开发过程中，软件工程师通常需要提供**容错模块和代码**。显然这会增加软件开发的**复杂度和冗余度**

软件缺陷的应对方法（2/2）

□发现缺陷

- ✓通过有效的**技术和管理手段来发现这些软件缺陷**
- ✓例如，制定和实施软件质量保证计划、开展软件文档和模型的评审、程序代码的走查、软件测试等工作。它们都可以帮助软件工程师找到潜藏在文档、模型和代码中的软件缺陷

□修复缺陷

- ✓通过一系列的手段来修复缺陷
- ✓采用程序调试等手段来**找到缺陷的原因、定位缺陷的位置**，进而**修改存在缺陷的程序代码**，将软件缺陷从软件制品中**移除出去**

2.4 软件缺陷的状态（1/2）

□尚未确认（Unconfirmed）

- ✓有人汇报了软件缺陷，但是尚未确认该软件缺陷是否真实存在

□有效（New）

- ✓经过确认，所汇报的软件缺陷真实存在，被正式视为是新缺陷，并等待进一步处理

□无效（Invalid）

- ✓经过确认，所汇报的软件缺陷并不存在，是一个无效的软件缺陷汇报

□重复（Duplicate）

- ✓该软件缺陷之前已经有人汇报过，属于重复性的软件缺陷

软件缺陷的状态（2/2）

□已分配（Assigned）

- ✓以安排人员负责修复缺陷

□已修复（Fixed）

- ✓缺陷已经修复

□信息不完整（Incomplete）

- ✓缺陷的描述信息不完整，导致无法准确和清晰地理解缺陷的内容

□已解决（Resolved）

- ✓针对该缺陷的处理已经完成

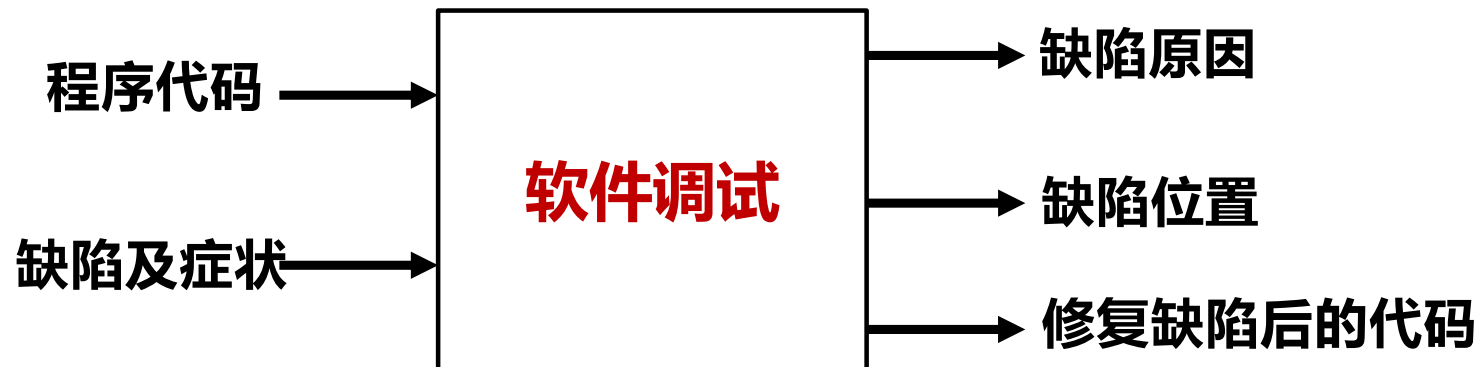
□已关闭（Closed）

- ✓关闭该缺陷，后续将不再针对该缺陷采用任何措施

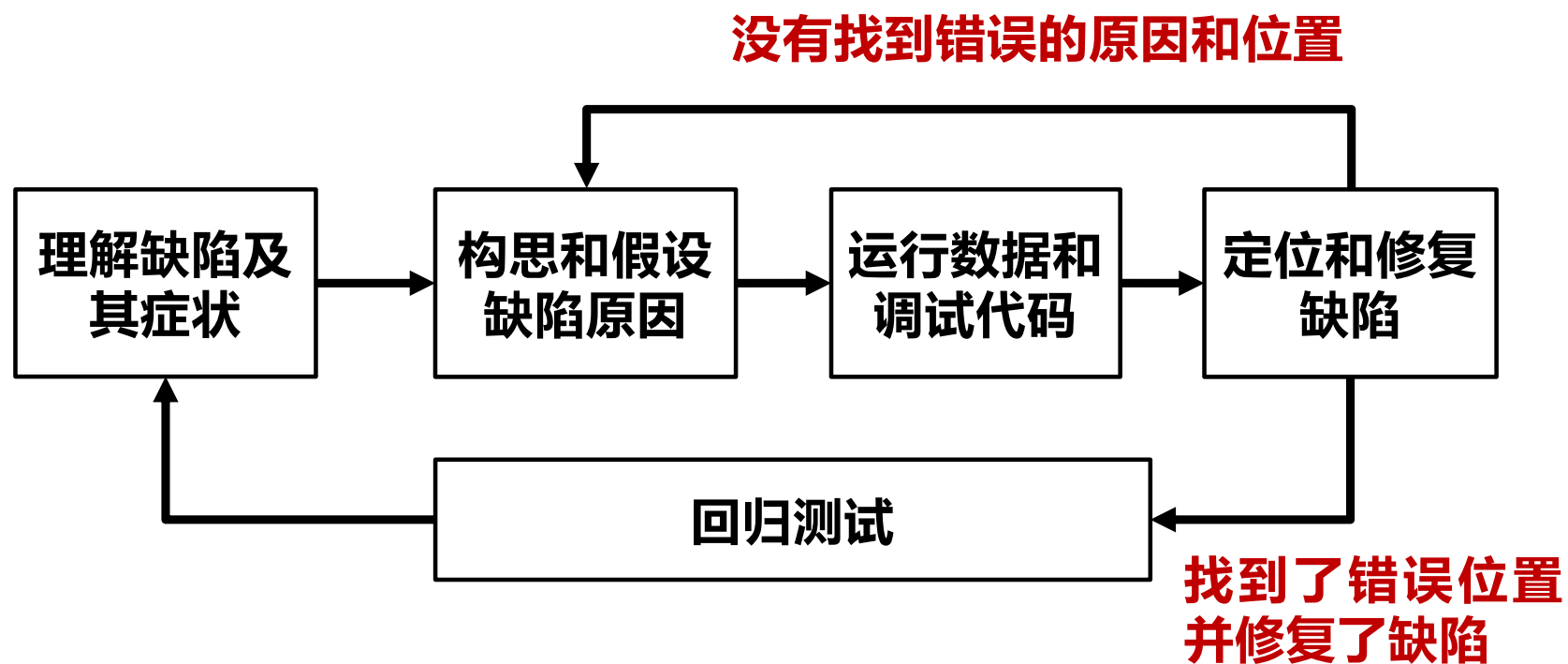
2.5 程序调试

□软件调试就是要基于程序代码，确定软件缺陷的原因、定位缺陷的位置，从而知道那里错了、如何修复缺陷

- ✓程序员需要花费大量的时间和精力用于软件调试
- ✓软件调试通过运行目标软件系统的程序代码，找到缺陷的代码位置、明确软件错误的具体原因，从而开展缺陷修复工作



调试的步骤



内容

1. 编写代码

- ✓任务、过程和方法
- ✓代码片段的重用

2. 软件缺陷和调试

- ✓软件缺陷、错误和失效
- ✓代码缺陷的应对方法及调试

3. 解决编程和调试问题

- ✓开源技术问答社区
- ✓群智知识的利用



3.1 编码和调试面临的挑战

□编码和调试需要开放的知识

- ✓包括软件设计的文档和模型、程序设计语言、程序调试技术等等

□编码和调试要求程序员有丰富的软件编程经验、扎实的编码和调试的技能、熟练的软件开发工具使用技巧等

□编程和调试中仍然会遇到各种各样的棘手问题

- ✓明明知道程序出现了错误，但是找不到错误的原因
- ✓程序中的错误有时会出现，有时候不会出现
- ✓程序代码和他人的程序代码一模一样，但是运行结果就是不正确

3.2 解决编程和调试问题的方法

- **独立自主解决问题**，但是有时候会出现无法解决的情况，用几个小时甚至几天的时间都未能解决问题
- **寻找团队成员的帮忙**，让有经验的编程高手帮助程序员解决问题
- **借助于开源技术问答社区中的互联网大众来解决问题**

技术问答社区

□ 常见社区

✓ CSDN、Stack Overflow

□ 交流形式

✓ 提出问题

✓ 回答问题

✓ 参加评论

最新 最热 悬赏 待回答 筛选

0 2
回答 浏览

matlab iris bp 看了你的，有些没懂
能加你个联系方式，问你一些问题嘛。是否可以请教一下您。有偿...

有问必答 matlab weixin_44192691 2021-09-02 15:59

stackoverflow Products Search...

Home PUBLIC Questions Tags Users COLLECTIVES Explore Collectives FIND A JOB Jobs Companies TEAMS

Top Questions

Ask Question

Interesting 306 Bountied Hot Week Month

5 votes 2 answers 2k views Facebook Login and email verification
django facebook oauth-2.0 facebook-login python-social-auth
answered 1 min ago code007 2,297

0 votes 3 answers 45 views Find string with maximum length of given attribute as string in java 8
java java-8
answered 1 min ago Bentaye 8,347

0 votes 0 answers 7 views How can I loop the Try block until no exceptions are caught?
java exception try-catch
asked 2 mins ago StockAZ89 29

0 votes 1 answer 16 views inner join in r gives much more rows than either data frame
r join dplyr
answered 2 mins ago Jon Spring 31.4k

海量的群体和群智知识

- 用户有1500万
- 问题有2200万
- 回答有3200万
- 评论有8200万

The screenshot displays the Stack Overflow website interface. On the left is a navigation sidebar with links for Home, PUBLIC (Questions, Tags, Users), COLLECTIVES (Explore Collectives), FIND A JOB (Jobs, Companies), and TEAMS. The main content area is titled 'Top Questions' and includes a search bar and an 'Ask Question' button. Below this are filters for Interesting (305), Bountied, Hot, Week, and Month. The list of questions includes:

- How can I use another image beside password toggle in Text input Layout In android Studio?**
0 votes, 0 answers, 2 views. Tags: java, xml, android-studio. Asked 50 secs ago by YourSelenophile 1.
- Space between SVG stroke and fill**
3 votes, 3 answers, 1k views. Tag: svg. Modified 1 min ago by Alexander Ulitin 1,104.
- Creating OTP page for SwiftUI Using TextField**
1 vote, 0 answers, 23 views. Tags: swiftui, uitablefield, one-time-password. Modified 1 min ago by Steve Vinoski 19k.
- testing whether property exists on an object and is equal to a certain value**
0 votes, 3 answers, 18 views. Tag: javascript. Answered 2 mins ago by Thomas 154k.

获取群智知识的方法

- 访问Stack Overflow、CSDN等社区
- 描述和输入自己遇到的问题
- 寻找针对该问题的有效解答

示例：查找的问题 “socket connection”

13,080 results

Relevance Newest More ▾

1824
votes

A: How do SO_REUSEADDR and SO_REUSEPORT differ?
A TCP/UDP **connection** is identified by a tuple of five values: {<protocol>, <src addr>, <src port>, <dest addr>, <dest port>}. Any unique combination of these values identifies a **connection**. ... Setting SO_REUSEADDR on a **socket** in Windows behaves like setting SO_REUSEPORT and SO_REUSEADDR on a **socket** in BSD, with one **exception**: Prior to Windows 2003, a **socket** with SO_REUSEADDR could always been ...

linux windows sockets unix portability

answered Jan 17 '13 by Mecki

17
votes

Q: Python socket connection exception
I have a **socket-connection** going on and I wanna improve the **exception** handling and Im stuck. ... Also, is it okay to exit programs using sys.exit when an unwanted **exception** occurs? Thank you ...

2
answers

python sockets exception exception-handling

asked Aug 22 '14 by erling

302
votes

Q: No connection could be made because the target machine actively refused it?
System.Net.WebException: Unable to connect to the remote server ---> System.Net.Sockets.SocketException: No **connection** could be made because the target machine actively refused it 127.0.0.1:80 at System.Net.Sockets.Socket.DoConnect ... s4, **Socket** s6, **Socket**& **socket**, IPAddress& address, ConnectSocketState state, IAsyncResult asyncResult, Int32 timeout, **Exception**& **exception**) --- End of inner **exception** stack trace --- at System.Net.HttpWebRequest.GetRequestStream ...

30
answers

c# .net asp.net-web-api2 socketexception system.net.webexception

asked Jun 4 '10 by hsnkvk

编写代码的输出

- 源程序代码
- 程序单元测试报告

小结

□编写代码

- ✓任务是要产生高质量程序代码，完成单元测试、程序调试等活动
- ✓基于软件设计模型和文档来编写代码
- ✓可以通过重用技术问答社区中的代码片段来编写程序

□软件缺陷、错误和失效

- ✓缺陷是指软件制品中不正确的描述和实现，缺陷的内在表现是程序运行产生不正确或者不期望的运行状态，导致程序无法为用户提供所需的功能和行为
- ✓调试目的是要发现缺陷原因、定位缺陷位置，促进缺陷的修复

□借助于技术问答社区来解决编码和和调试中遇到的问题

思考和讨论

你是如何解决编码中遇到的问题？借助群智的方法来解决编程问题可信和可行吗？



问题和讨论

