

**成都信息工程大学区块链产业学院**

**《面向对象程序设计（Java）A》**

**上机实验报告**

|  |  |
| --- | --- |
| **实验课程** | **面向对象程序设计（Java）A** |
| **实验项目** | **实验四 Java面向对象高级编程实验** |
| **学生姓名** | **熊灵欣** |
| **学生学号** | **2021131126** |
| **班 级** | **区块链工程214班** |
| **指导教师** | **李 凡** |
| **实验地点** | **6610实验室** |
| **完成日期** | **2021年12月7日** |

## 实验名称

实验四 Java面向对象高级编程实验

## 实验目的

通过本实验，学生学习掌握使用Java面向对象高级程序设计相关知识，综合运用类的继承、抽象类、super关键字、接口等知识点，设计开发验证性程序。

1. 学习使用Eclipse IDE调试Java程序
2. 进一步掌握类的继承的程序设计方法
3. 掌握super关键字的使用方法
4. 进一步掌握抽象类在系统设计中的使用方法
5. 掌握接口在系统设计中的使用方法

## 实验内容

1. SHA256算法是SHA2安全哈希算法的一种，对于长度小于264位（bit）的输入数据，SHA256的哈希值结果长度为256位（bit），即32字节。SHA256处理每组数据块大小为512位（bit），内部的处理单元位数为32位（bit），通过64次迭代运算，最终生成哈希值结果。SHA256算法是比特币系统所采用的哈希算法。Java的java.security包中的MessageDigest类提供了SHA256哈希算法的功能实现，设计类HashCalculater，该类的构造方法以字符串作为参数，并提供calculateHash()方法，返回该字符串作为输入计算生成的哈希值对应的16进制数字符串。

【示例程序】

**package experiment.four;**

**import java.math.BigInteger;**

**import java.security.MessageDigest;**

**import java.security.NoSuchAlgorithmException;**

**import java.util.Scanner;**

**public class HashCalculator {**

**private String inputMsg;**

**public HashCalculator(String s) {**

**inputMsg=s;**

**}**

**public String caculateHash() {**

**try {**

**byte[] bytes = inputMsg.getBytes();**

**MessageDigest md = MessageDigest.*getInstance*("SHA-256");**

**md.reset();**

**byte[] hash = md.digest(bytes);**

**if (hash.length != 32) {**

**System.*out*.println("计算hash值失败");**

**return null;**

**}**

**BigInteger bInt=new BigInteger(hash);**

**return bInt.toString(16);**

**} catch (NoSuchAlgorithmException e) {**

**e.printStackTrace();**

**}**

**return null;**

**}**

**public static void main(String[] args) {**

**Scanner cmd=new Scanner(System.*in*);**

**System.*out*.println("输入任意的字符串：");**

**String msg=cmd.nextLine();**

**HashCalculator hashC=new HashCalculator(msg);**

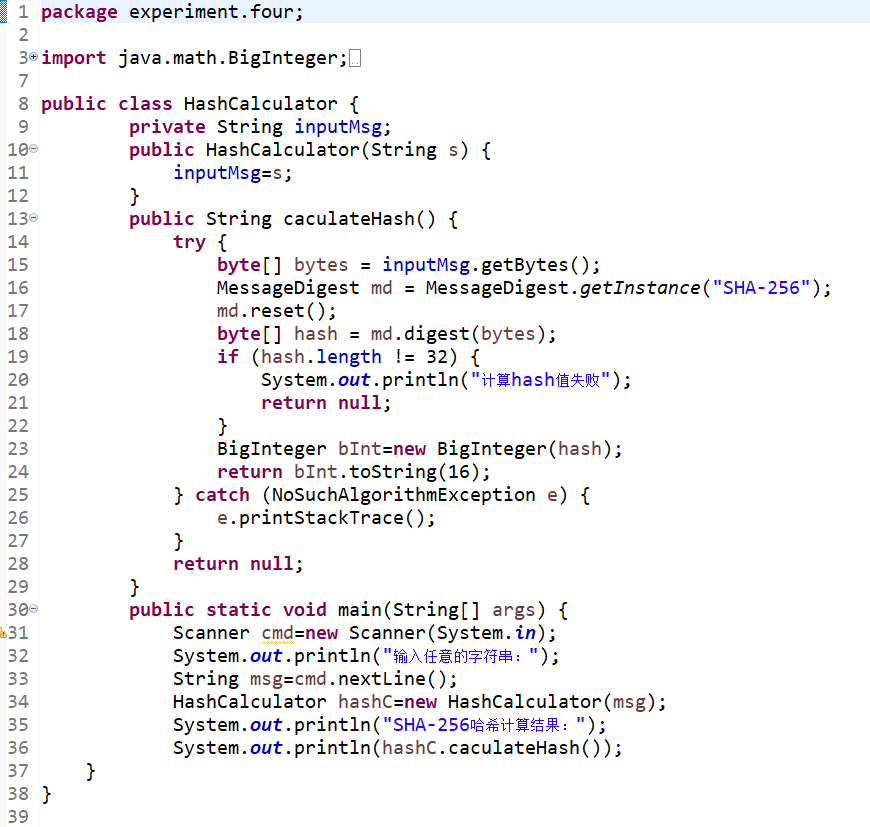
**System.*out*.println("SHA-256哈希计算结果：");**

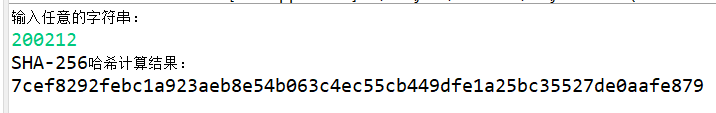
**System.*out*.println(hashC.caculateHash());**

**}**

**}**

使用Eclipse IDE编写上面的程序，并编译执行，观看运行结果，并附上源程序代码截图和程序执行结果的屏幕截图。





1. 已知表示区块链中的一个区块的Java类Block，区块包括序号、交易信息、哈希结果、前一个区块的哈希结果、创建时间、难度值等属性，分别使用Block类对象的变量 (long index，String data，String hash，String previousHash，Date timestamp，int difficulty）进行定义，根据程序框架完成类Block的代码。

【程序框架】

**package experiment.four;**

**import java.util.Date;**

**public class Block {**

**private long index;**

**private String hash;**

**private String previousHash;**

**private String data;**

**private Date timeStamp;**

**private int difficulty;**

**public Block(long i,String data,String hash,String pHash,int d) {**

...缺失代码...

**this.timeStamp = new Date();**

**}**

**public long getIndex() {**

...缺失代码...

**}**

**public String getHash() {**

...缺失代码...

**}**

**public String getPreviousHash() {**

...缺失代码...

**}**

**public String getData() {**

...缺失代码...

**}**

**public Date getTimestamp() {**

...缺失代码...

**}**

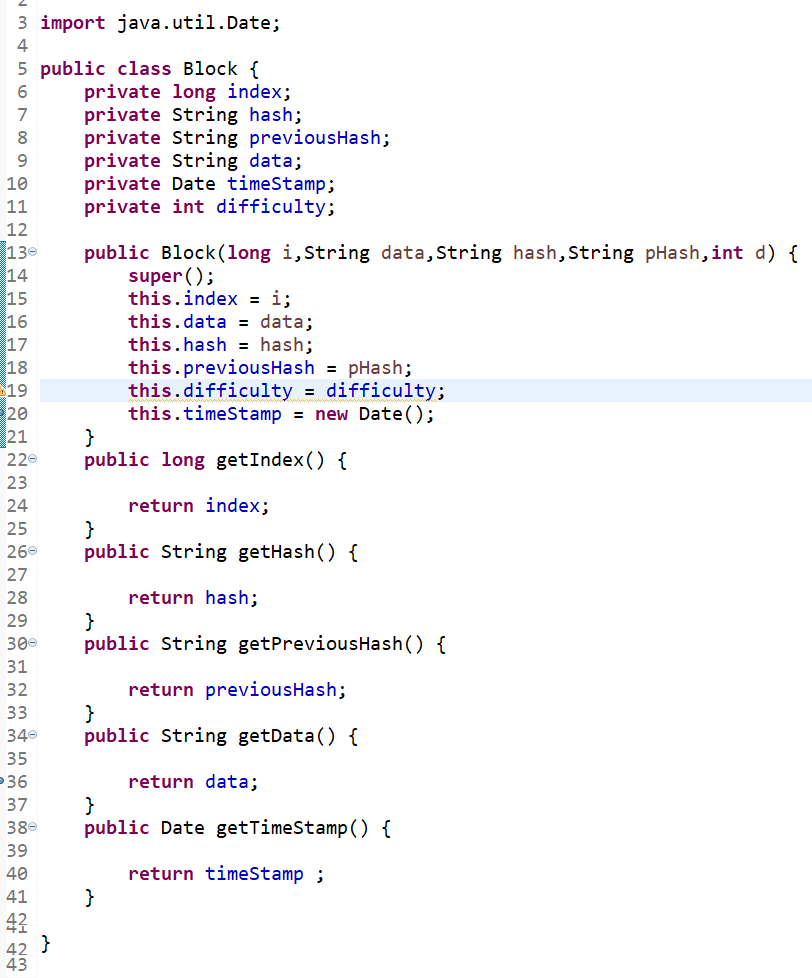
**public int getDifficulty() {**

...缺失代码...

**}**

**}**

使用Eclipse IDE编写上面的程序，补充缺失的程序代码，实现程序的功能要求，附上源程序代码截图。



1. 编写程序模拟客人在猫咖里给猫科和狗类动物喂养各种不同食物的过程。当客人给动物喂食时，动物发出不同欢快的叫声。在猫咖里，涉及的对象有客人，各种不同猫科和狗类动物以及各种不同的食物。使用类Customer表示客人，抽象类Animal表示所有的动物，Animal的子类Cat和Dog表示猫科和狗类动物，接口Edible表示可以吃的食物。

* **Customer类**的示例代码如下。

**public class Customer {**

**private String name;**

**public Customer (String name) {**

**this.name = name;**

**}**

**public void speak(Animal a) {**

**if( ...缺失代码... )**

**System.out.println("这只小猫\""+a.getName()+"\"真可爱！");**

**if( ...缺失代码... )**

**System.out.println("这只小狗\""+a.getName()+"\"真可爱！");**

**}**

**public void feed(Animal a, Edible food) {**

**a.eat(food);**

**}**

**public String getName() {**

**return name;**

**}**

**public void setName(String name) {**

**this.name = name;**

**}**

**}**

* **抽象类Animal**的示例代码如下。

**public abstract class Animal {**

**private String name;**

**public Animal(String name) {**

**this.name = name;**

**}**

**public abstract void shout();**

**public abstract void eat(Edible food);**

**public String getName() {**

**return name;**

**}**

**public void setName(String name) {**

**this.name = name;**

**}**

**}**

* Animal的子类**Cat类**的示例代码如下。

**public class Cat extends Animal{**

**public Cat(String name) {**

**...缺失代码...**

**}**

**public void shout() {**

**System.out.println("喵喵.....");**

**}**

**public void eat(Edible f) {**

**System.out.println("小猫 " +getName() + " 正在吃着 " + f.getName());**

**}**

**}**

* Animal的子类**Dog类**的示例代码如下。

**public class Dog extends Animal {**

**public Dog(String name) {**

**...缺失代码...**

**}**

**public void shout() {**

**System.out.println("汪汪.....");**

**}**

**public void eat(Food f) {**

**System.out.println("小狗 " +getName() + " 正在吃着 " + f.getName());**

**}**

**}**

* 接口Edible的示例代码如下。

**public interface Edible {**

**String getName();**

**}**

* 猫科动物食物Fish类的示例代码如下。

**public class Fish implements Food{**

**private String name="鱼肉";**

**public String getName() {**

**return name;**

**}**

**}**

* 狗类动物食物Bone类的示例代码如下。

**public class Bone implements Food{**

**private String name="骨头";**

**public String getName() {**

**return name;**

**}**

**}**

* 已知测试主程序代码如下。

**public class Test {**

**public static void main(String[] args) {**

**Customer c1 = new Customer ("张三");**

**Customer c2 = new Customer ("李四");**

**Dog dog = new Dog("二哈");**

**Edible f1 = new Bone();**

**c1.speak(dog);**

**c1.feed(dog, f1);**

**Cat cat = new Cat("柯南");**

**Edible f2 =new Fish();**

**c2.speak(cat, f2);**

**c2.feed(cat, f2);**

**}**

**}**

使用Eclipse IDE编写上面的程序，补充缺失的程序代码，实现程序的功能要求，附上源程序代码截图。

