

**计算机与信息 学院实验报告**

|  |  |  |
| --- | --- | --- |
| 实验课程： | 面向对象程序设计实验 | |
| 实验编号： | 实验四 | |
| 实验名称： | 类的封装性（2） | |
| 实验人员： | 学号 | 22111302041 |
| 姓名 | 高蝶 |
| 班级 | 计算机科学与技术 |
| 指导教师： | 陶涛 | |
| 实验室： | 学院楼2号楼202 | |
| 实验日期： | 2023.3.14 | |

# 实验目的

1. 充分掌握面向对象编程规范及面向对象编程的一般方法

2. 理解类成员访问权限的限定及相关规则；

3. 利用面向对象程序思想设计较复杂编程问题的解决方案，并在设计时体现出创新意识。

# 实验要求

1. 所编写的程序应符合类的封装性要求，符合面向对象程序编写规范；

2. 能利用封装性进行复杂程序的设计和实现；

3. 对问题进行充分的需求分析，设计正确、全面的测试数据，设计类的定义框架及UML图后，再进行编程实现；

4. 所设计的测试数据全部通过后，再进行程序提交，提高程序的通过率。

# 实验内容

**1.验证型实验**

**1.1成员的访问权限**

阅读以下程序，分析其中类的定义和方法的定义，并进行上机验证，使程序能够运行并得到结果。如果程序有错，请在“五、实验结果与分析”处分析原因，提出改正方案（使用自然语言表达清楚错误部分，不要粘贴完整源码），并给出Order类中main方法的运行结果。**注意：不要修文件名、类名和包名。**

//UserTest.java

package user;

public class UserTest {

public static void main(String[] args) {

User u1=new User();

System.out.println(u1.login());

u1.userID="101";

System.out.println(u1.login());

}

}

//user.java

package user;

class User{

String userID="";

String userPassword;

boolean login() {

if(userID.equals("101") && userPassword.equals("666666"))return true;

else return false;

}

}

//Order.java

package order;

class Order {

public static void main(String[] args) {

User u=new User();

System.out.println(u.login());

}

}

**1.2 基本类型的封装**

以下Java程序的功能是：输入一行字符串，其中包含数字字符和其他字符，将其中的数字串提取后，转化成整数，并计算这些整数的和。仔细分析下面的Java语言源程序，分析程序中的错误，并将程序补充完整，在“五、实验结果与分析”处分析错误原因，提出改正方案和补充的程序代码（使用自然语言表达清楚错误部分，不需要粘贴完整源码）。给出三组测试数据，并进行验证。

import java.util.Scanner;

public class ExtractNumber {

public static void main(String[] args){

int i,sum;

Scanner reader=new Scanner(System.in);

String str=reader.nextLine();//输入一行字符串，包括空格，此行无错误

String digit[]=str.split("\\D");// “\\D”表示非数字字符，split方法的作用是以非数字字符作为分隔符，对字符串进行分隔，此行无错误

System.out.println(digit.length);//输出分隔后字符串数组的长度

for(i=0;i<digit.length;i++) {

//请在此处将程序补充完整

}

System.out.println("sum="+sum);

}

}

提示：split方法对字符串进行分隔后，可能会产生空串（没有内容的空串""，不是null）。

**2.应用型实验**

**2.1类的封装**

请定义一个名为Fan的类模拟风扇，其属性包括打开状态、速度、风扇半径、风扇颜色。要求为每个属性提供访问器和修改器（getter和setter），定义print方法输出风扇对象的相关属性值。风扇有四种种固定的速度，分别是静止（quiet，0）慢速（slow，1-3）、中速（middle，4-7）、快速（quick，8-9）；定义测试类，输入速度、半径、颜色、打开状态等数据，按格式输出相关信息。

相关属性值约束：

（1）打开状态：ON或者OFF，表示打开、关闭；

（2）速度小于0时赋值0，速度不可以大于9（大于9时赋值9）；

（3）风扇半径在[5,10]内合法，大于10的赋值10，小于5的赋值5；

（4）风扇颜色只有黑色（black）、灰色（gray）、白色（white）三种，其他输入均默认为黑色（black）；

（5）打开状态为OFF时，速度无效（即无论输入什么，都应赋值0）打开状态为ON时，速度也可以是0；

（6）私有属性、公有方法

注意：本题重点考查程序是否符合面向对象编程规范，包括变量命名、属性定义、getter和setter方法、成员方法定义、方法调用、代码对齐方式，等等。本题会在实验报告中重点检查。

输入：风扇的打开状态、速度、风扇半径、风扇颜色等相关属性值，需要验证输入数据的合法性（已知数据类型正确，只需验证数值是否合法）；每次输入n组数据。

输出：n组风扇的各属性信息

提示：

（1）测试类（主类）结构参考“实验一设计型实验（1）分数计算”；

（2）不需要定义对象数组，按以下方式处理n组数据问题：

Fan fan=new Fan();

for（i=0;i<n;i++）

{

fan.setStatus(reader.next());

fan.setSpeed(reader.nextInt());

……

fan.print();

}

（3）变量、方法定义见名知意，getter和setter方法由eclipse自动生成后，再进行修改。

样例输入：

4

ON 1 8 black

OFF 3 11 red

ON 10 5 gray

OFF 0 3 white

样例输出：

Status:ON;Speed:slow;FanRadius:8;FanColor:black

Status:OFF;Speed:quiet;FanRadius:10;FanColor:black

Status:ON;Speed:quick;FanRadius:5;FanColor:gray

Status:OFF;Speed:quiet;FanRadius:5;FanColor:white

程序框架参考：

import java.util.Scanner;

public class FanTest {

public static void main(String[] args) {

int i,n;

Scanner reader=new Scanner(System.in);

n=reader.nextInt();

Fan fan;

for(i=0;i<n;i++) {

fan=new Fan();

fan.setStatus(reader.next());

fan.setSpeed(reader.nextInt());

fan.setRadius(reader.nextInt());

fan.setColor(reader.next());

fan.print();

}

}

}

class Fan{

//成员变量定义

public String getStatus() {

}

public void setStatus(String status) {

}

public int getSpeed() {

}

public void setSpeed(int speed) {

}

public int getRadius() {

}

public void setRadius(int radius) {

}

public String getColor() {

return color;

}

public void setColor(String color) {

}

public String getSpeedType() {

//根据速度，判断速度类型

return speedType;

}

public void print() {

//按格式要求输出相关信息

}

}

**2.2 提取身份证信息**

身份证号码中各数位的意义为：

1-2位：省、自治区、直辖市代码；

3-4位：地级市、盟、自治州代码；

5-6位：县、县级市、区代码；

7-14位：出生年月日，比如19670401代表1967年4月1日；

15-17位：为顺序号，其中17位男为单数，女为双数；

18位：为校验码，0-9和X。作为尾号的校验码，是由号码编制单位按统一的公式计算出来的，如果某人的尾号是0－9，都不会出现X，但如果尾号是10，那么就得用X来代替，因为如果用10做尾号，那么此人的身份证就变成了19位。X是罗马数字的10，用X来代替10。有的地方也允许使用小写从x。

身份证校验码的计算方法

1、将前面的身份证号码17位数分别乘以不同的系数。从第一位到第十七位的系数分别为：7－9－10－5－8－4－2－1－6－3－7－9－10－5－8－4－2。

2、将这17位数字和系数相乘的结果相加。

3、用加出来和除以11，看余数是多少？

4、余数只可能有0－1－2－3－4－5－6－7－8－9－10这11个数字。其分别对应的最后一位身份证的号码为1－0－X－9－8－7－6－5－4－3－2。其中，小写x也认为正确。

5、通过上面得知如果余数是3，就会在身份证的第18位数字上出现的是9。如果对应的数字是2，身份证的最后一位号码就是罗马数字X。

请你编写程序，验证所输入身份证号的合法性，如果合法，则输出其年龄（周岁，根据出生日期判断年龄）。

输入：当前日期，n及个模拟身份证号，n组

输出：合法性（true,false），年龄（整数）

样例输入：

2022 6 14

3

330903198509060219

340101200010113224

330329200504255448

样例输出：

true:36

false:-1

true:17

以下程序框架和相关方法的调用情况，供参考。可依据程序框架添加、完善相关方法代码。

import java.util.Scanner;

public class IDCard {

public static void main(String[] args) {

Scanner reader=new Scanner(System.in);

int y,m,d,n,i;

y=;

m=;

d=;

n=;

IdentityCard card;

for(i=0;i<n;i++) {

card=new IdentityCard();

card.setID(reader.next());

if(card.check()==true)

System.out.println("true:"+card.getAge(y,m,d));

else

System.out.println("false:-1");

}

}

}

class IdentityCard{

//定义成员变量，getter,setter方法

public boolean check() {

if(!ID.matches("[1-9][0-9]{5}(19|20)[0-9]{9}[0-9Xx]")) {//一个简易的正则表达式，判断身份证号前17位是否为数字，第一位不能是0，第7/8位是19或20，最后一位是数字或X、x，本语句可以直接使用。

return false;

}

year=Integer.parseInt(ID.substring(6, 10)); //获取身份证号码中的出生年份

month=Integer.parseInt(ID.substring(10, 12)); //获取身份证号码中的出生月份

day=Integer.parseInt(ID.substring(12, 14)); //获取身份证号码中的出生日

if(!dateLegal()) {//判断出生日期是否合法，主要是月、日的合法性

return false;

}

char ch[]=ID.toCharArray();//将字符串转换成字符数组

//根据定义，计算加权总和sum

yu=sum%11;

//根据余数对应关系，判断身份证号最后一位是否正确

}

boolean dateLegal(){ //判断出生日期是否合法

}

public int getAge(int y,int m,int d) {

//根据当前日期和生日，计算年龄

}

}

**3. 设计型实验**

（1）共饮同井水（升级版）

某村庄有一口公共水井，为了保护该水井及饮用水的质量，当地规定当该水井内的水量不高于100时，禁止取水6小时整（假设1:0:0开始禁止取水，则7:0:0可解禁），在此期间，公共水井水量自动恢复到最大容量1000；如果预计村民当次取水后水量低于100，则拒绝该村民本次取水。

其实每个村民自家也有口水井，自家的水井容量各不相同，范围在(100,200]内。村民优先去公共水井取水，若发现公共水井禁止取水或因容量不够拒绝取水时，再从自家水井取水。

自家水井容量不高于100时，禁止取水1小时整，在此期间，水井水量自动恢复到初始水量；如果预计村民当次取水后水量低于100，则允许该村民本次从自家水井取水至水井容量到100。

请模拟n次村民的用水情况：先设置公共水井水量及每位村民自家水井容量，再模拟村民取水的情形。村民每次取水时需输入年、月、日、时、分、秒和取水量，若成功从公共水井取水则输出“public:m”形式，m表示公共水井剩余水量；若成功从自家水井取水则输出“private:n”形式，n表示自家水井剩余水量；若没有取到水，则输出“wait”。

输入：公共水井初始水量（100<publicWell<=1000）；m及m位村民自家水井水量（100<privateWell<=200，也是初始水量，村名编号从1开始，按1递增）；n及n次村民的取水需求，每次取水格式：村民编号 年 月 日 时 分 秒 实际取水量(>0)。（不用检查输入数据格式的规范性，后一次取水时间晚于前一次取水时间，年份大于1970）

输出：每次取水的情况，显示方式（村民编号:水井类型,成功取水量,本次取水后水井的剩余水量）：

样例输入1：

1000

3 150 200 110

4

1 2022 3 1 18 25 30 800

2 2022 3 1 20 25 20 150

1 2022 3 1 23 25 19 100

3 2022 3 2 6 25 25 150

样例输出1：

1:public,800,200

2:private,100,100

1:public,100,100

3:public,150,850

样例输入2：

101

5 125 150 175 200 200

11

1 2020 1 23 12 30 30 1

2 2020 2 21 12 23 23 100

3 2020 2 28 23 24 21 800

4 2020 3 1 4 4 4 800

5 2020 12 31 23 24 25 100

1 2021 1 1 3 4 5 21

3 2021 2 28 23 12 23 100

4 2022 2 28 23 45 58 800

1 2022 3 1 4 5 56 100

3 2023 2 28 23 59 59 2000

3 2023 3 1 0 1 1 901

样例输出2：

1:public,1,100

2:public,100,900

3:public,800,100

4:public,800,200

5:public,100,100

1:private,21,104

3:public,100,900

4:public,800,100

1:private,4,100

3:private,75,100

3:wait

提示：

（1）定义一个类DateTime，处理日期、时间的间隔，或者利用Calender类处理日期的间隔；

（2）将公共水井取水分成两类：取水成功、取水失败；将私人水井分成三种情况：取全部水、取部分水、取水失败；

（3）将小时转换成秒进行比较。

# 程序清单

2.应用型实验

（1）

测试数据

第1组：4

ON 1 8 black

OFF 3 11 red

ON 10 5 gray

OFF 0 3 white

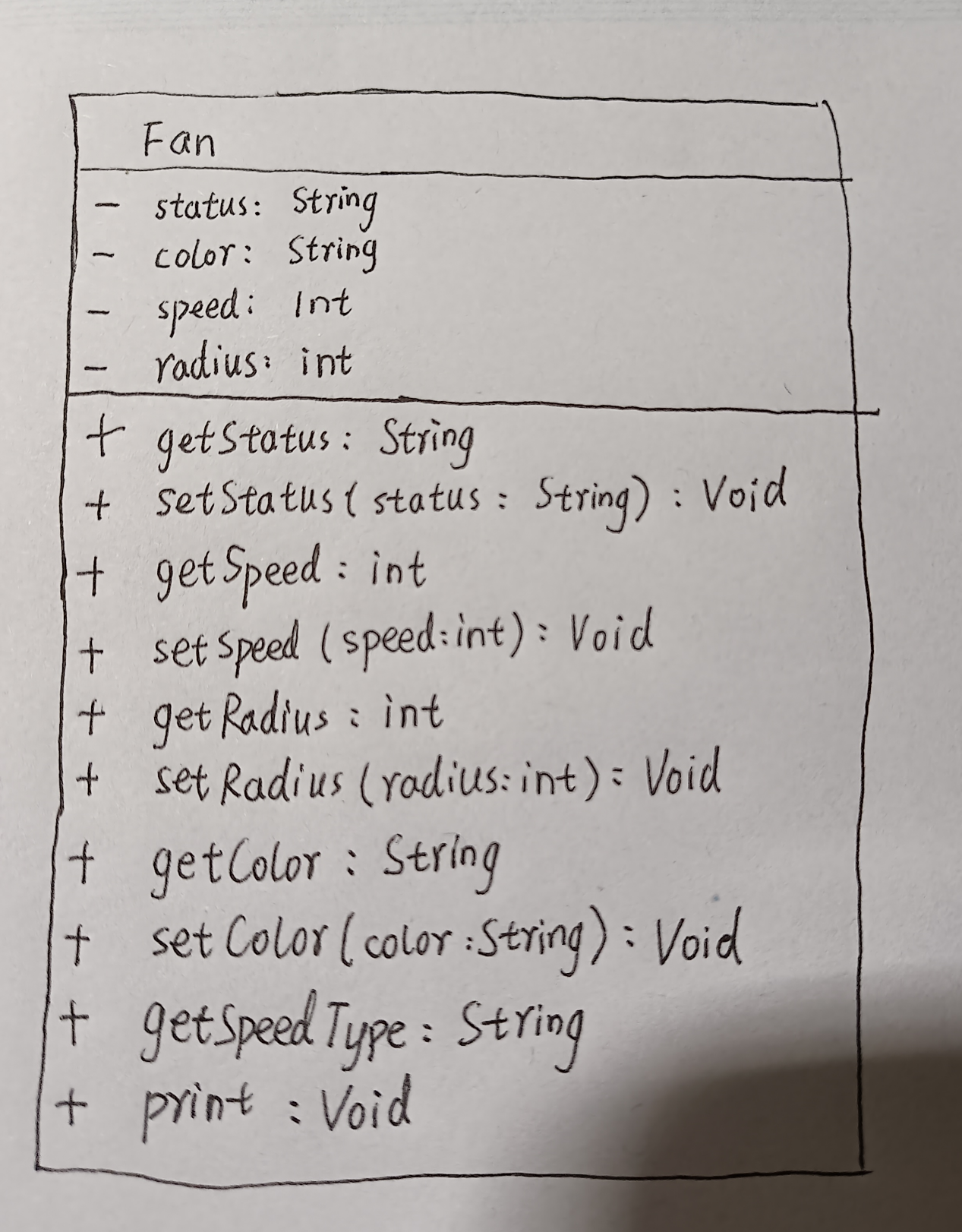
第2组：2

ON 0 3 red

OFF 1 8 gray

第3组：1

ON 3 10 black

类的UML图：

程序代码：**package** prj\_fanmoni;

**import** java.util.Scanner;

**public** **class** Main {

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

**int** i,n;

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

n=reader.nextInt();

Fan fan;

**for**(i=0;i<n;i++) {

fan=**new** Fan();

fan.setStatus(reader.next());

fan.setSpeed(reader.nextInt());

fan.setRadius(reader.nextInt());

fan.setColor(reader.next());

fan.print();

}

}

}

**class** Fan{

**private** String status,color;

**private** **int** speed,radius;

**public** String getStatus() {

**return** status;

}

**public** **void** setStatus(String status) {

**this**.status=status;

}

**public** **int** getSpeed() {

**return** speed;

}

**public** **void** setSpeed(**int** speed) {

**if**(status.equals("OFF")) {

**this**.speed=0;

}**else**

{

**if**(speed<0) {

**this**.speed=0;

}**else** **if**(speed>9) {

**this**.speed=9;

}**else** {

**this**.speed=speed;

}

}

}

**public** **int** getRadius() {

**return** radius;

}

**public** **void** setRadius(**int** radius) {

**if**(radius>10) {

**this**.radius=10;

}**else** **if**(radius<5) {

**this**.radius=5;

}**else** {

**this**.radius=radius;

}

}

**public** String getColor() {

**return** color;

}

**public** **void** setColor(String color) {

**if**(color.equals("black")||color.equals("gray")||color.equals("white")) {

**this**.color=color;

}**else** **this**.color="black";

}

**public** String getSpeedType() {

String speedType = **null**;

**if**(speed==0) {

speedType="quiet";

}**else** **if**(speed>=1&&speed<=3) {

speedType="slow";

}**else** **if**(speed>=4&&speed<=7) {

speedType="middle";

}**else** **if**(speed>=8&&speed<=9) {

speedType="quick";

}

**return** speedType;

}

**public** **void** print() {

System.***out***.println("Status:"+status+";"+"Speed:"+getSpeedType()+";"+"FanRadius:"+radius+";FanColor:"+color);

}

}

（2）

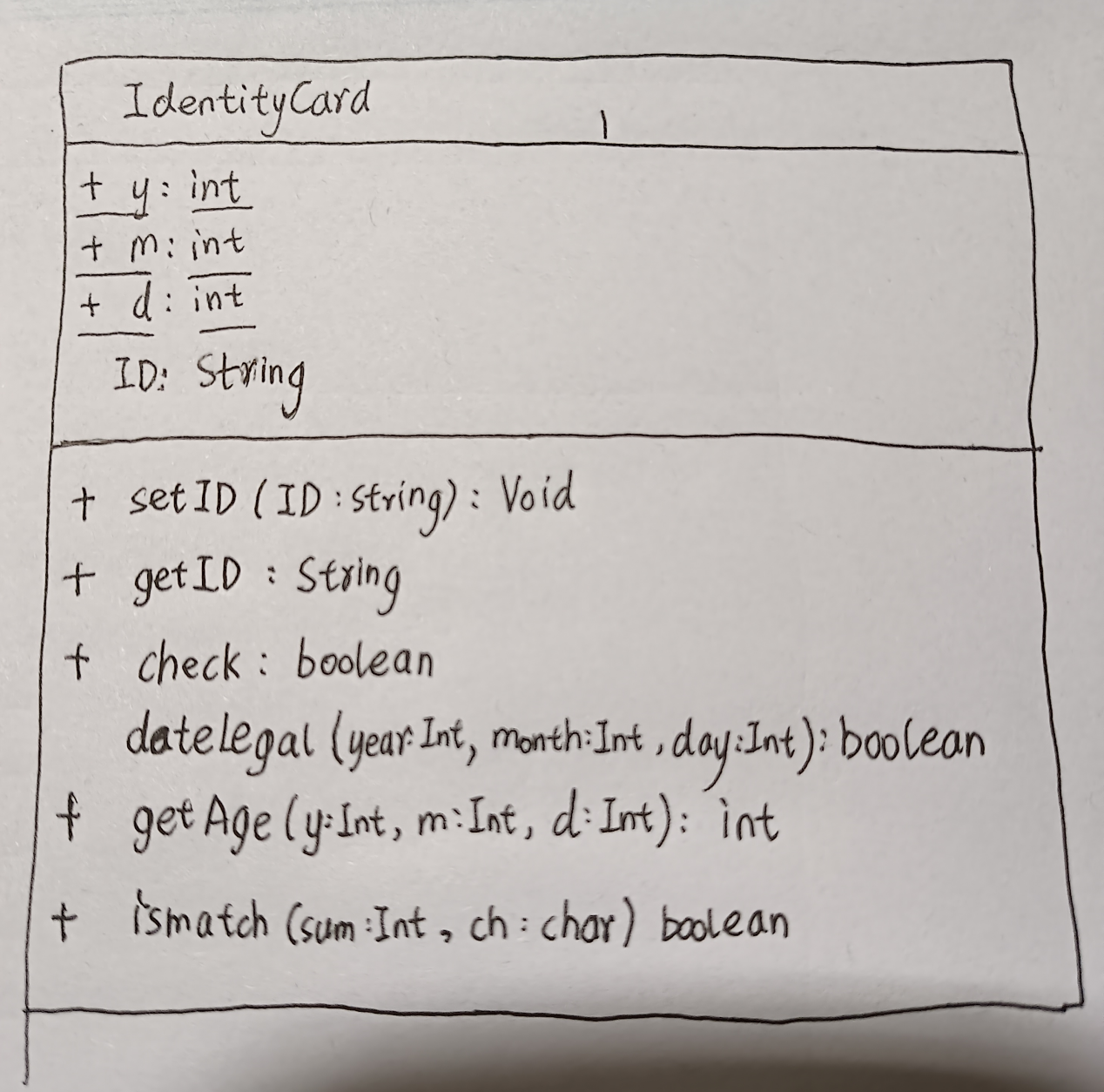
测试数据

第1组：2022 6 14 3

330903198509060219

340101200010113224

330329200504255448

类的UML图：

程序代码：**package** prj\_getidinfo;

**import** java.util.Scanner;

**public** **class** Main {

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

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

**int** y,m,d,n,i;

y=reader.nextInt();

IdentityCard.*y*=y;

m=reader.nextInt();

IdentityCard.*m*=m;

d=reader.nextInt();

IdentityCard.*d*=d;

n=reader.nextInt();

IdentityCard card;

**for**(i=0;i<n;i++) {

card=**new** IdentityCard();

card.setID(reader.next());

**if**(card.check()==**true**)

System.***out***.println("true:"+card.getAge(y,m,d));

**else**

System.***out***.println("false:-1");

}

}

}

**class** IdentityCard{

**public** **static** **int** *y*,*m*,*d*;

String ID=**null**;//定义成员变量，getter,setter方法

**public** **void** setID(String ID) {

**this**.ID=ID;

}

**public** String getID() {

**return** ID;

}

**public** **boolean** check() {

**int** year,month,day,yu,sum=0;

**int**[] number= {7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2};

**if**(!ID.matches("[1-9][0-9]{5}(19|20)[0-9]{9}[0-9Xx]")) {

//一个简易的正则表达式，判断身份证号前17位是否为数字，第一位不能是0，第7/8位是19或20，最后一位是数字或X、x，本语句可以直接使用。

**return** **false**;

}

year=Integer.*parseInt*(ID.substring(6, 10)); //获取身份证号码中的出生年份

month=Integer.*parseInt*(ID.substring(10, 12)); //获取身份证号码中的出生月份

day=Integer.*parseInt*(ID.substring(12, 14)); //获取身份证号码中的出生日

**if**(!dateLegal(year,month,day)) {

//判断出生日期是否合法，主要是月、日的合法性

**return** **false**;

}

**char** ch[]=ID.toCharArray();//将字符串转换成字符数组

**for**(**int** i=0;i<17;i++) {

sum=sum+number[i]\*(ch[i]-'0');

}

yu=sum%11;

// System.out.println(yu);

**if**(ismatch(yu,ch[17])) **return** **true**; //根据余数对应关系，判断身份证号最后一位是否正确

**else** **return** **false**;

}

**boolean** dateLegal(**int** year,**int** month,**int** day ){ //判断出生日期是否合法

**int**[]a= {31,28,31,30,31,30,31,31,30,31,30,31};

**if**(year%400==0||year%4==0&&year%100!=0) a[1]=29;

**if**(year>*y*) **return** **false**;

**else** **if**(year==*y*&&month>*m*) **return** **false**;

**else** **if**(year==*y*&&month==*m*&&day>day) **return** **false**;

**else** **if**(year<=0)**return** **false**;

**else** **if**(month>12||month<=0) **return** **false**;

**else** **if**(day>a[month-1]||day<=0) **return** **false**;

**else** **return** **true**;

}

**public** **int** getAge(**int** y,**int** m,**int** d) {

**int** year=Integer.*parseInt*(ID.substring(6, 10));

**int** month=Integer.*parseInt*(ID.substring(10, 12));

**int** day=Integer.*parseInt*(ID.substring(12, 14));

**if**(m>month||m==month&&d>=day)**return** y-year;

**else** **return** y-year-1;

}

**public** **boolean** ismatch(**int** sum,**char** ch) {

**if**(sum!=2) {

**if**(sum+(ch-'0')==12||sum+(ch-'0')==1) **return** **true**;

**else** **return** **false**;

}**else** {

**if**(ch=='x'||ch=='X') **return** **true**;

**else** **return** **false**;

}

}

}

3.设计型实验

（1）

测试数据

第1组：1000

3 150 200 110

4

1 2022 3 1 18 25 30 800

2 2022 3 1 20 25 20 150

1 2022 3 1 23 25 19 100

3 2022 3 2 6 25 25 150

第2组：800

2

150 200

1

2 2023 3 1 3 20 30 400

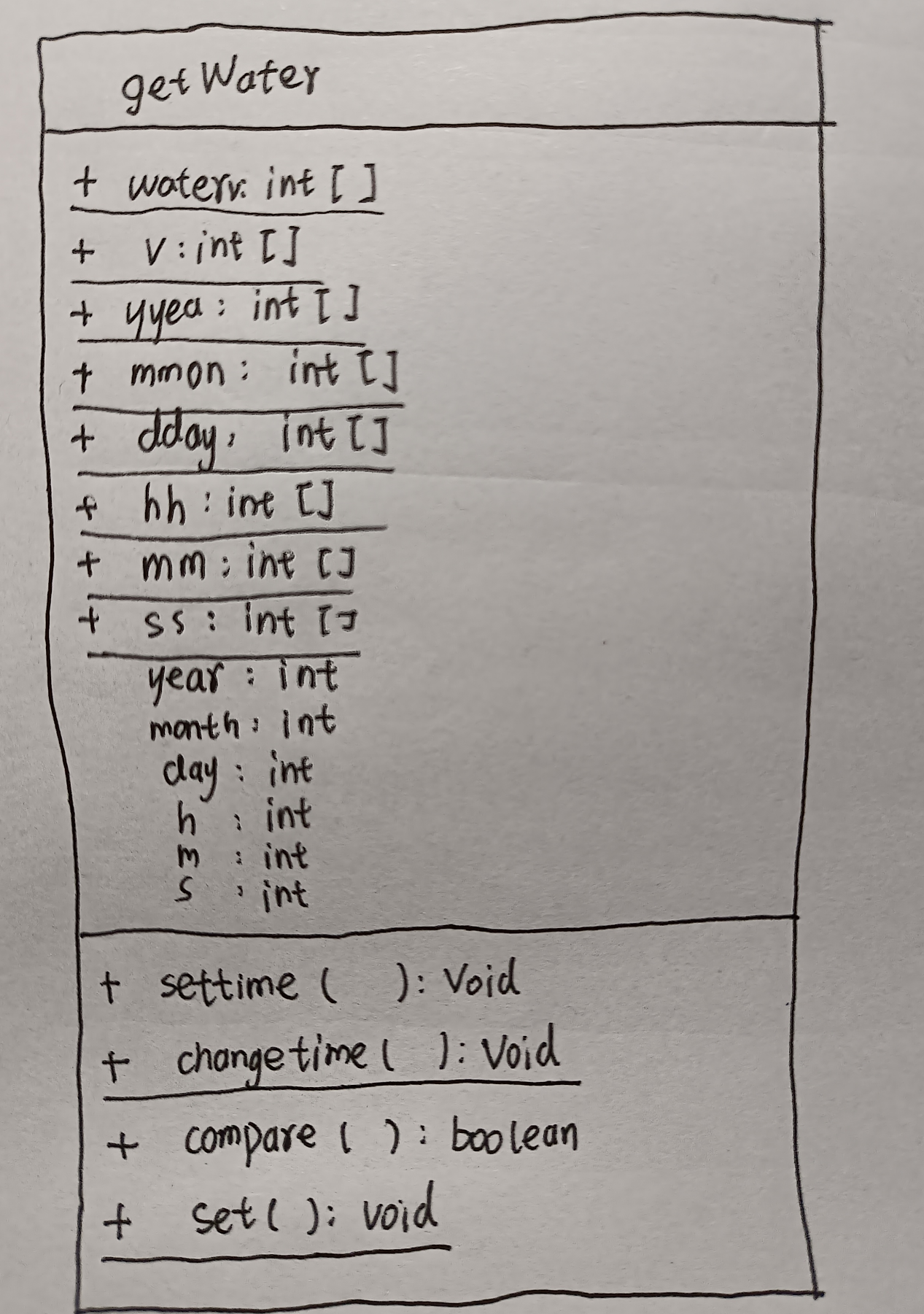
第3组：800

2 150 200

2

1 2022 3 4 12 30 23 500

1 2022 3 4 1 30 23 300

类的UML图：

程序代码：package prj\_drinkwater2;

import java.util.Scanner;

import java.util.Calendar;

public class Main {

public static Scanner reader=new Scanner(System.in);

public static void main(String[] args) {

int waterv[]=new int[5000],n,m;

waterv[0]=reader.nextInt();

m=reader.nextInt();

for(int i=1;i<=m;i++ ){

waterv[i]=reader.nextInt();

}

n=reader.nextInt();

getWater p[]=new getWater[n+10];

getWater.getwater(waterv,m);

getWater.set(p,n);

}

}

class getWater{

public static int waterv[]=new int [5000],v[]=new int [5000],yyear[]=new int [5000],mmon[]=new int [5000],dday[]=new int [5000],hh[]=new int [5000],mm[]=new int [5000],ss[]=new int [5000];

int year,month,day,h,m,s,index;

public static void getwater(int[] watervv,int m){

for(int i=0;i<=m;i++) {

waterv[i]=watervv[i];

v[i]=watervv[i];

}

}

public void settime(int year,int month,int day,int h,int m,int s) {

this.year=year;

this.month=month;

this.day=day;

this.h=h;

this.m=m;

this.s=s;

}

public static void changetime(getWater p,int i) {

yyear[i]=p.year;mmon[i]=p.month;dday[i]=p.day;hh[i]=p.h;mm[i]=p.m;ss[i]=p.s;

int t[]= {31,28,31,30,31,30,31,31,30,31,30,31};

if(yyear[i]%400==0||yyear[i]%4==0&&yyear[i]%100!=0) {

t[1]=29;

}

if(i==0)hh[i]+=6;

else hh[i]+=1;

if(hh[i]>=24) {

hh[i]%=24;

dday[i]+=1;

if(dday[i]>t[mmon[i]-1]) {

dday[i]=1;

mmon[i]++;

if(mmon[i]>12) {

mmon[i]=1;

yyear[i]++;

}

}

}

}

public boolean compare(int i) {

// if(year>yyear[i]) return true;

// else if(year==yyear[i]&&month>mmon[i]) return true;

// else if(year==yyear[i]&&month==mmon[i]&&day>dday[i])return true;

// else if(year==yyear[i]&&month==mmon[i]&&day==dday[i]&&h>hh[i]) return true;

// else if(year==yyear[i]&&month==mmon[i]&&day==dday[i]&&h==hh[i]&&m>mm[i])return true;

// else if(year==yyear[i]&&month==mmon[i]&&day==dday[i]&&h==hh[i]&&m==mm[i]&&s>=ss[i]) return true;

// else return false;

Calendar dateTime = Calendar.getInstance();

dateTime.set(year,month-1, day, h, m,s);

long time = dateTime.getTimeInMillis()/1000;

Calendar dateTime2 = Calendar.getInstance();

dateTime2.set(yyear[i],mmon[i]-1,dday[i],hh[i],mm[i],ss[i]);

long timei = dateTime2.getTimeInMillis()/1000;

if(time>=timei) return true;

else return false;

}

public static void set(getWater[] a,int n) {

int water,y,mo,d,h,m,s,index;

for(int i=0;i<n;i++) {

Scanner reader=Main.reader;

a[i]=new getWater();

index=reader.nextInt();

y=reader.nextInt();

mo=reader.nextInt();

d=reader.nextInt();

h=reader.nextInt();

m=reader.nextInt();

s=reader.nextInt();

a[i].settime(y,mo,d,h,m,s);

water=reader.nextInt();

if(a[i].compare(0)) {

if(waterv[0]-water>100){

System.out.printf("%d:public,%d,%d\n",index,water,waterv[0]-water);

waterv[0]-=water;

}else if(waterv[0]-water==100){

System.out.printf("%d:public,%d,%d\n",index,water,waterv[0]-water);

getWater.changetime(a[i],0);

waterv[0]=1000;

}

else {

if(a[i].compare(index)){

if(waterv[index]-water>100){

System.out.printf("%d:private,%d,%d\n",index,water,waterv[index]-water);

waterv[index]-=water;

}else if(waterv[index]-water==100){

System.out.printf("%d:private,%d,%d\n",index,water,waterv[index]-water);

getWater.changetime(a[i],index);//gai

waterv[index]=v[index];

}

else {

System.out.printf("%d:private,%d,%d\n",index,waterv[index]-100,100);

getWater.changetime(a[i],index);//gai

waterv[index]=v[index];

}

}else {

System.out.println(index+":"+"wait");

}

}

}else {

if(a[i].compare(index)) {

if(waterv[index]-water>100) {

System.out.printf("%d:private,%d,%d\n",index,water,waterv[index]-water);

waterv[index]-=water;

}else if(waterv[index]-water==100) {

System.out.printf("%d:private,%d,%d\n",index,water,waterv[index]-water);

getWater.changetime(a[i],index);//gai

waterv[index]=v[index];

}

else {

System.out.printf("%d:private,%d,%d\n",index,waterv[index]-100,100);

getWater.changetime(a[i],index);//gai

waterv[index]=v[index];

}

}else {

System.out.println(index+":"+"wait");

}

}

}

}

}

# 实验结果及分析

1.验证型实验

（1）错误：在不同包中调用默认方法。应在调用的那个方法前加上public.

运行结果：false

false

1. 错误：sum未初始化，应改为sum=0；

代码补充：**if**(digit[i].equals("")==**false**)

sum+=Integer.*parseInt*(digit[i]);

测试数据：12,,1,2

1,1,2,3,4,

23;12;cv2;3

运行结果：4

sum=15

5

sum=11

7

sum=40

2.应用型实验

（1）1：Status:ON;Speed:slow;FanRadius:8;FanColor:black

Status:OFF;Speed:quiet;FanRadius:10;FanColor:black

Status:ON;Speed:quick;FanRadius:5;FanColor:gray

Status:OFF;Speed:quiet;FanRadius:5;FanColor:white

2：Status:ON;Speed:quiet;FanRadius:5;FanColor:black

Status:OFF;Speed:quiet;FanRadius:8;FanColor:gray

3：Status:ON;Speed:slow;FanRadius:10;FanColor:black

（2）true:36

false:-1

true:17

3.设计型实验

（1）1：1:public,800,200

2:private,100,100

1:public,100,100

3:public,150,850

2：2:public,400,400

3：1:public,500,300

1:private,50,100

# 实验小结

[本次实验的反思、收获和体会]

反思：1：有些功能的实现尽量用java提供的类来实现，如时间计算比较可以用Calendar类，自己写的代码难免避免不了许多小细节考虑不到，也可以去查该类的源码学习学习。

2：面向对象编程规范性不强，代码还有很多面向过程的痕迹。

收获：1：对类成员访问权限的限定及相关规则有了进一步的熟悉与了解，能够一点点使用了。

2：面向对象编程规范性进一步强化，了解了这样做的必要性。

3：面对复杂的现实问题能够耐着性子慢慢看，一点点想如何实现。

体会：代码的逻辑上的错误真的很难找出来，自己考虑问题也确实不全面，debug的能力有待加强。认真读题理解题意，题目小细节很多真的要反复揣摩。看了同学们的代码，既能从中学到一些自己没用过的类以及写法以拓宽解题思路，也能从中发现一些常犯的错误加深对一些知识点的理解。