Java实验报告

@*Author* : *liyajun*_208111545116

实验一

搭建环境并输出*Hello World*

```
public static void main(String[] args){
    // write your code here
    System.out.println("Hello World");
}
```

```
 .jar=55238:C:\Users\liyaj\AppData\Local\Jet
 demo.demo01
Hello World

Process finished with exit code 0
```

实验二

回文数

H5 代码实现

```
 boolean valid(int i) {
    String s = Integer.toString(i);
    int l = 0,r = s.length() - 1;
    while(l < r) {
        if(s.toCharArray()[l++] != s.toCharArray()[r--])
return false;
    }
    return true;
}
```

H5 结果

```
12121 -> true;
114514 -> false;
```

猜数字

H5 代码实现

```
public void guess(){
    Random ra = new Random();
    int cur = ra.nextInt(100) + 1;
    System.out.println("please enter a num");
    Scanner sc = new Scanner(System.in);
    int get = sc.nextInt();
    while(get != cur){
        if(get > cur) System.out.println("Try a smaller
number");
        else if(get < cur) System.out.println("Try a larger
number");
        get = sc.nextInt();
    }
    System.out.println("congratulations");
}
```

H5 结果

```
 .jar=56111:C:\Users\liyaj\AppData\Local\JetBrains\Toolbox
 demo.demo01
please enter a num
50
Try a larger number
85
Try a smaller number
65
Try a larger number
75
congratulations

Process finished with exit code 0
```

完数

H5 代码实现

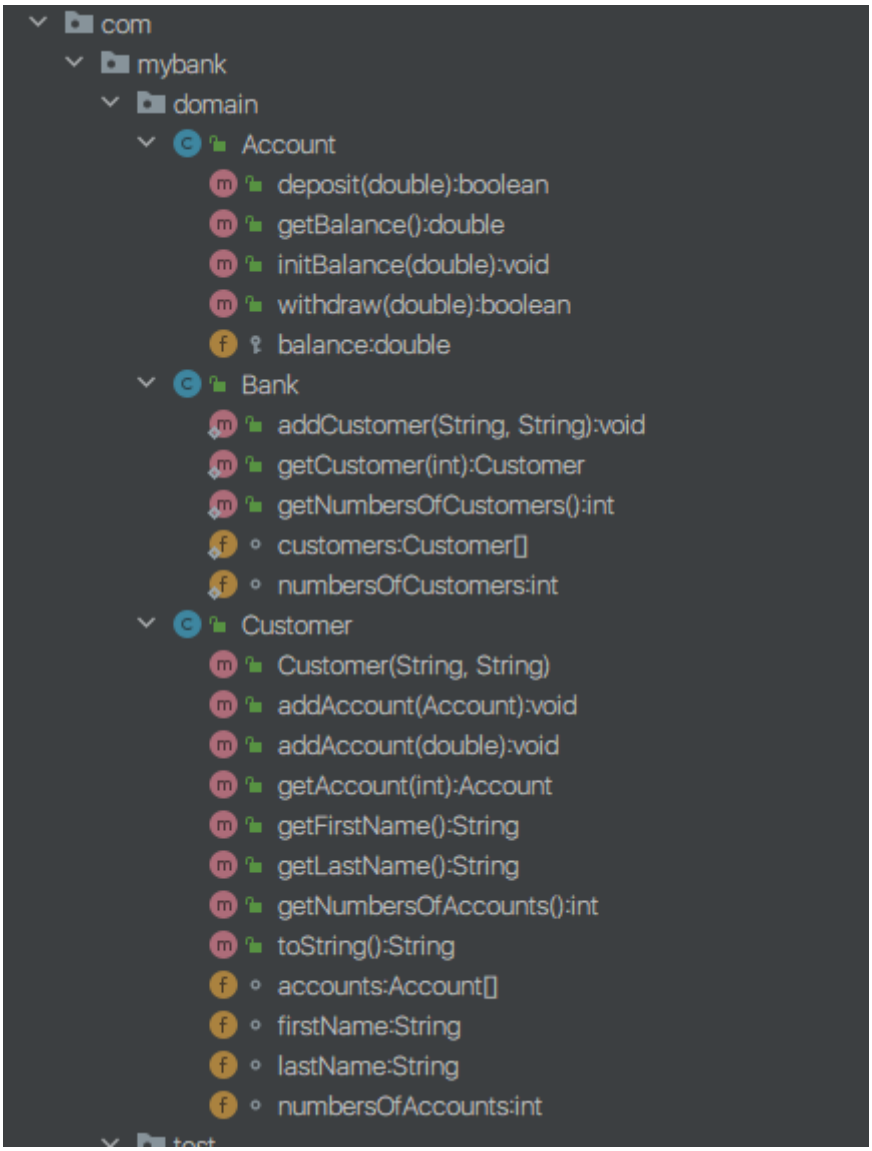
```java
public static void main(String[] args){
    // write your code here
    for(int i = 1;i <= 1000; i++){
        if(i % 2 == 0 && valid(i)) {
            System.out.println(i);
        }
    }
}
boolean valid(int x) {
    int cur = x;
    for(int i = 1; i <= x/2; i++) {
        if(x % i == 0) cur -= i;
    }
    return cur == 0;
}
```

H5 运行结果

```
6
28
496
```

实验三

项目结构



实现代码

H5 Account.java

```java
package com.mybank.domain;

public class Account {
    protected double balance;
   //初始化余额
    public Account(double balance){
```

```
        this.balance = balance;
    }
    //存钱
    public boolean deposit(double amt){
        if(amt < 0) return false;
        balance += amt;
        return true;
    }
    //取钱
    public boolean withdraw(double amt){
        if(balance < amt) return false;
        balance -= amt;
        return true;
    }

    //获取余额
    public double getBalance(){
        return balance;
    }
}

```

Bank.java

```
package com.mybank.domain;

public class Bank {
    static Customer [] customers;
    static int numbersOfCustomers;
    static {
        customers = new Customer[10];
        numbersOfCustomers = 0;
    }
    public static void addCustomer(String firstName, String
lastName){
        Customer cus = new Customer(firstName,lastName);
        try {
            customers[numbersOfCustomers++] = cus;
        }catch (ArrayIndexOutOfBoundsException e){
            System.out.println(e);
        }
    }

```

```
    public static int getNumbersOfCustomers(){
        return numbersOfCustomers;
    }
    public static Customer getCustomer(int id){
        try {
            return customers[id];
        } catch (ArrayIndexOutOfBoundsException e){
            System.out.println(e);
        }
        return null;
    }
}

```

H5 **Customer.java**

```
package com.mybank.domain;

public class Customer {
    String firstName;
    String lastName;
    Account [] accounts;
    int numbersOfAccounts;

    public Customer(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
        numbersOfAccounts = 0;
        accounts = new Account[10];
    }
    public void addAccount(double amount){
           addAccount(new Account(amount));
    }
    public void addAccount(Account acc){
        try {
            accounts[numbersOfAccounts++] = acc;
        }catch (ArrayIndexOutOfBoundsException e){
            System.out.println(e);
        }
    }
    public Account getAccount(int id){
        try {

```

```
            return accounts[id];
        }catch (ArrayIndexOutOfBoundsException e){
            System.out.println(e);
        }
        return  null;
    }
    public int getNumbersOfAccounts() {
        return numbersOfAccounts;
    }
    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }

    @Override
    public String toString() {
        return  firstName + "-" +  lastName ;
    }
}

```

TestBanking.java

```
package com.mybank.test;

import com.mybank.domain.Bank;

import java.util.Random;

public class TestBanking {
    public static void main(String[] args) {
        //初始化成员
        for(int i = 0; i < 10; i++){
            Random ra = new Random();
            Bank.addCustomer(Integer.toString(i), "Jinx");
            for(int j = 0;j < 10; j++){

 Bank.getCustomer(i).addAccount(ra.nextDouble(1145.14));


```

```
15
 Bank.getCustomer(i).getAccount(j).withdraw(ra.nextDouble(1145
.14));
16              }
17          }
18          for(int i = 0;i < 10; i++){
19
 System.out.println(Bank.getCustomer(i).toString());
20              for(int j = 0;j < 10; j++){
21                  System.out.println("第"+(j + 1 )+"个账户的余额
为");
22
 System.out.println(String.format("%.2f",Bank.getCustomer(i).
getAccount(j).getBalance()));
23              }
24          }
25
26      }
27  }
28
```

测试结果

```
C:\Users\liyaj\.jdks\openjdk-17.0.2\bin\java.exe -javaagent:C:\Users\liyaj\AppData\Local\JetBrains\Toolbox\apps\IDEA-U\ch
  .jar=57769:C:\Users\liyaj\AppData\Local\JetBrains\Toolbox\apps\IDEA-U\ch-0\213.7172.25\bin -Dfile.encoding=UTF-8 -classp
  com.mybank.test.TestBanking
0-Jinx
第1个账户的余额为
72.67
第2个账户的余额为
184.98
第3个账户的余额为
30.48
第4个账户的余额为
57.66
第5个账户的余额为
721.69
第6个账户的余额为
191.60
第7个账户的余额为
764.83
第8个账户的余额为
454.07
第9个账户的余额为
223.04
第10个账户的余额为
22.55
1-Jinx
第1个账户的余额为
320.37
第2个账户的余额为
625.96
第3个账户的余额为
23.68
第4个账户的余额为
187.50
第5个账户的余额为
123.28
第6个账户的余额为
700.84
第7个账户的余额为
42.48
```

实验四

带有存款和贷款的银行系统

H5 **CheakAccount.java**

```java
package experiment.four.com.mybank.domain;
//能够透支的账户
public class CheckingAccount extends Account{
    double overdraftAmount;

    public CheckingAccount(double initBalance, double
overdraftAmount){
        super(initBalance);
        this.overdraftAmount = overdraftAmount;
    }
    CheckingAccount(double initBalance){
        super(initBalance);
```

```
        this.overdraftAmount = 0.0;
    }

    public double getOverdraftAmount() {
        return overdraftAmount;
    }

    @Override
    public int withdraw(double amount) {
        if(balance < amount){
            //使用贷款
            double overdraftNeeded = amount - balance;
            if(overdraftNeeded > overdraftAmount){
                return 1;
            }
            else{
                balance = 0.0;
                overdraftAmount -= overdraftNeeded;
                return 2;
            }
        } // 能够付清
        else{
            balance -= amount;
        }
        return 3;
    }
}

```

SavingAccount.java

```
package experiment.four.com.mybank.domain;
//能计算利息的账户
public class SavingsAccount extends Account{
    private double interestRate;
    public SavingsAccount(double initBalance,double
interestRate){
        super(initBalance);
        this.interestRate = interestRate;
    }

```

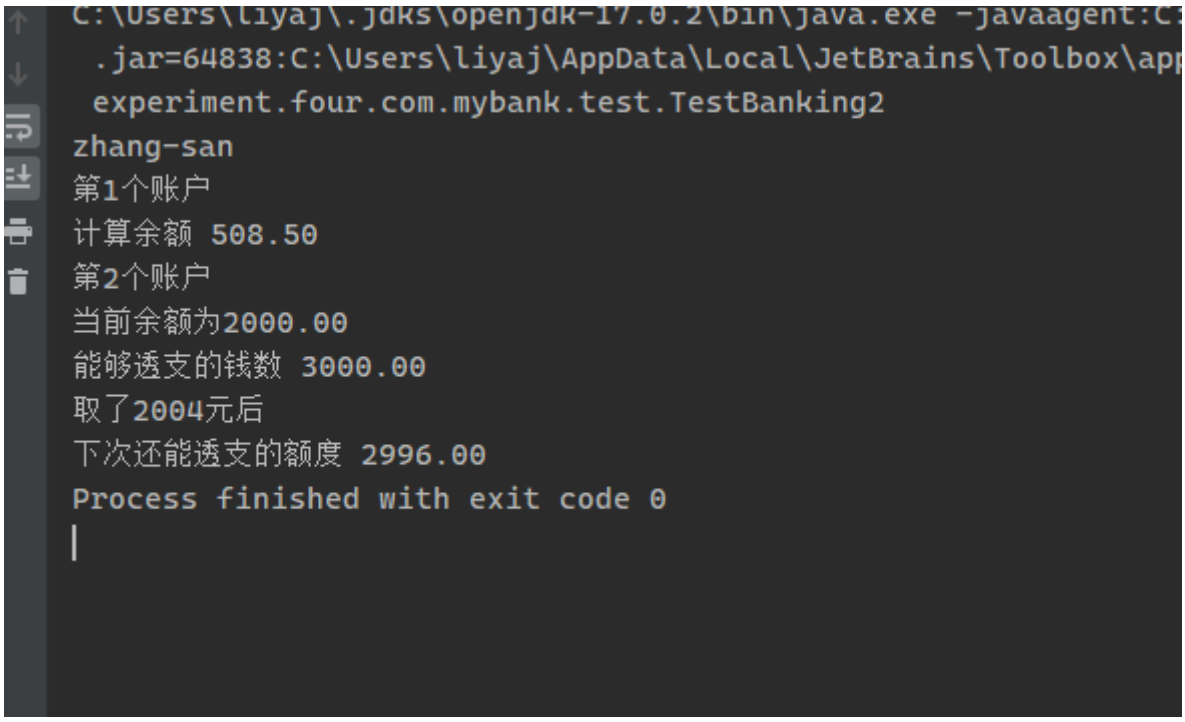
```
9
10
11      //计算利息
12      public double accumulateInterest(){
13          return balance + balance * interestRate;
14      }
15  }
16
```

H5 Test

```
1   package experiment.four.com.mybank.test;
2
3
4   import experiment.four.com.mybank.domain.*;
5
6
7   public class TestBanking2 {
8       public static void main(String[] args) {
9           Bank.addCustomer("zhang", "san");
10          Customer c1 = Bank.getCustomer(0);
11          c1.addAccount(new SavingsAccount(500,0.017));
12          c1.addAccount(new CheckingAccount(2000, 3000));
13          for(int i = 0;i < Bank.getNumbersOfCustomers();i++) {
14              Customer cus = Bank.getCustomer(i);
15              System.out.println(cus);
16              for(int j = 0;j < cus.getNumbersOfAccounts();j++)
    {
17                  Account acc = cus.getAccount(j);
18                  System.out.println("第"+(j+1)+"个账户");
19                  if(acc instanceof SavingsAccount sa){
20                      System.out.printf("计算余额
    %.2f\n",sa.accumulateInterest());
21                  }
22                  else if(acc instanceof CheckingAccount ck){
23                      System.out.printf("当前余额
    为%.2f\n",ck.getBalance());
24                      System.out.printf("能够透支的钱数
    %.2f\n",ck.getOverdraftAmount());
25                      ck.withdraw(2004);
26                      System.out.println("取了2004元后");
```

```
27                      System.out.printf("下次还能透支的额度
%.2f",ck.getOverdraftAmount());
28                  }
29                  else {
30                      //普通账户
31                      System.out.printf("余额为
%.2f",acc.getBalance());
32                  }
33              }
34          }
35      }
36  }
37
```

测试



实验五

动物宠物程序

实现代码

H5 Animal.java

```java
package experiment.five.InterfaceProject;

public abstract class Animal {
    protected int legs;
    protected  Animal(int legs){
        this.legs = legs;
    }
    abstract void eat();
    public void walk(){
        System.out.println("Animal walk with"+legs+"legs.");
    }
}

```

H5 Cat.java

```java
package experiment.five.InterfaceProject;

public class Cat extends Animal implements Pet{
    private String name;
    public Cat(String name){
        super(4);
        this.name = name;
    }
    public Cat(){
        this("");
    }
    @Override
    void eat() {
        System.out.println("Cat likes eating mice.");
    }

    @Override
    public String getName() {

```

```
        return this.name;
    }

    @Override
    public void setName(String n) {
        this.name = name;
    }

    @Override
    public void play() {
        System.out.println("Cat likes playing strings.");
    }
}
```

Duck.java

```
package experiment.five.InterfaceProject;

public class Duck extends Animal implements Pet,Sound{
    private String name;

    public Duck( String name) {
        super(2);
        this.name = name;
    }

    @Override
    public void walk() {
        System.out.println("duck swim and walks
with"+legs+"legs");
    }

    @Override
    void eat() {
        System.out.println("duck like eating waterweeds");
    }

    @Override
    public String getName() {
        return this.name;
    }

```

```
    @Override
    public void setName(String n) {
        this.name = name;
    }

    @Override
    public void play() {
        System.out.println("duck likes playing with water");
    }
    public void shout(String voice){
        Sound.super.shout(voice);
        System.out.println("i am a yellow duck");
    }
}
```

H5 Fish.java

```
package experiment.five.InterfaceProject;

public class Fish extends Animal implements Pet{
    private String name;
    @Override
    void eat() {
        System.out.println("Fish likes eating bugs in the
ponds");
    }

    public Fish() {
        super(0);
    }

    @Override
    public String getName() {
        return this.name;
    }

    @Override
    public void setName(String n) {
        this.name = n;
    }

    @Override
```

```
25      public void play() {
26          System.out.println("fish likes swimming in yje
    ponds");
27      }
28      @Override
29      public void walk(){
30          super.walk();
31          System.out.println("fish are swimming happily in the
    pool.");
32      }
33  }
```

H5 Pet.interface

```
1   package experiment.five.InterfaceProject;
2
3   public interface Pet {
4       public String getName();
5       public void setName(String n);
6       public void play();
7   }
```

H5 Sound.interface

```
1   package experiment.five.InterfaceProject;
2
3   public interface Sound {
4       public default void shout(String voice){
5           System.out.println(voice);
6       }
7   }
```

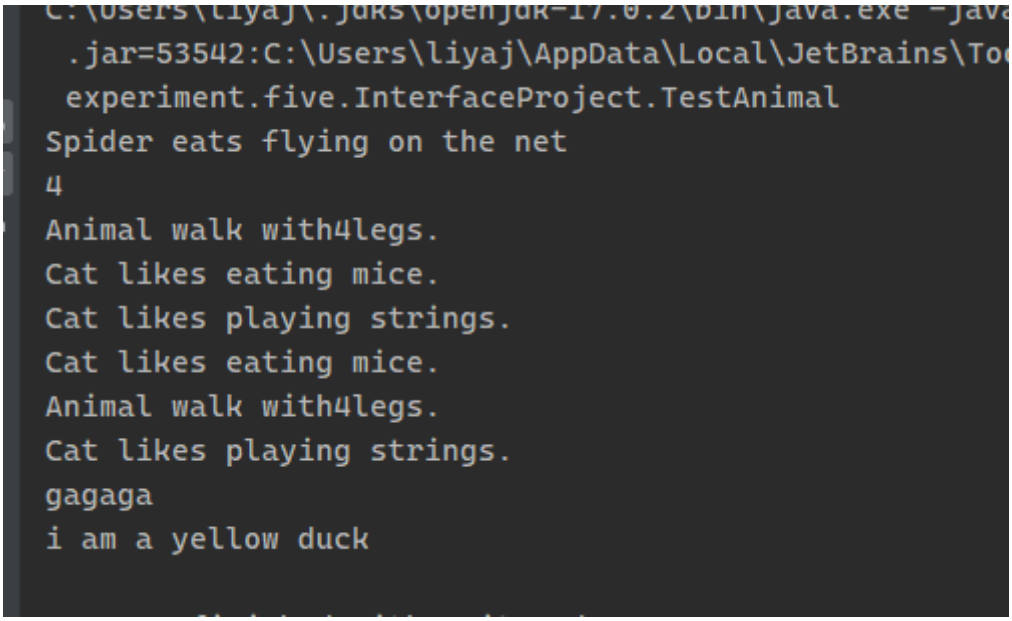
H5 Spider.java

```java
package experiment.five.InterfaceProject;

public class Spider extends Animal{
    public Spider(){
        super(8);
    }

    @Override
    void eat() {
        System.out.println("Spider eats flying on the net");
    }
}
```

H5 Test

```java
package experiment.five.InterfaceProject;

public class TestAnimal {
    public static void main(String[] args) {
        Spider s1 = new Spider();
        s1.eat();
        Animal s2 = new Cat();
        System.out.println(s2.legs);
        s2.walk();
        s2.eat();
        Pet s3 = new Cat("fluffy");
        s3.getName();
        s3.play();
        Cat s4 = new Cat("Grafield");
        s4.getName();
        int legs = s4.legs;
        s4.eat();
        s4.walk();
        s4.play();
        Pet s5 = new Fish();
        s5.getName();
        Duck s6  = new Duck("donald duck");
        s6.shout("gagaga");
    }
}
```

测试



实验六

实验代码

H5 容器*Bank*

```
1  package experiment.six.com.mybank.domain;
2
3  import java.util.ArrayList;
4  import java.util.List;
5
6  public class Bank {
7      static List<Customer> customers;
8      static {
9          customers = new ArrayList<>();
10     }
11     public static void addCustomer(String firstName, String
   lastName){
12         customers.add(new Customer(firstName,lastName));
13     }
14     public static int getNumbersOfCustomers(){
```

```
        return customers.size();
    }
    public static Customer getCustomer(int id){
        try {
            return customers.get(id);
        } catch (ArrayIndexOutOfBoundsException e){
            System.out.println(e);
        }
        return null;
    }
}
```

H5 容器*Customer*

```
package experiment.six.com.mybank.domain;

import java.util.ArrayList;
import java.util.List;

public class Customer {
    String firstName;
    String lastName;
    List<Account> accounts;
    //Account[] accounts;
    int numbersOfAccounts;

    public Customer(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
        accounts = new ArrayList<>();
    }
    public void addAccount(double amount){
           addAccount(new Account(amount));
    }
    public void addAccount(Account acc){
            accounts.add(acc);
    }
    public Account getAccount(int id){
        try {
            return accounts.get(id);
        }catch (ArrayIndexOutOfBoundsException e){
            System.out.println(e.getMessage());

```

```
            return null;
        }
    }
    public int getNumbersOfAccounts() {
        return accounts.size();
    }
    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }

    @Override
    public String toString() {
        return  firstName + "-" +  lastName ;
    }
}
```

H5 异常类*OverdraftException*

```
package experiment.six.com.mybank.domain;

/**
 * @author liyajun
 * @date 2022/4/6 14:40
 */
public class OverdraftException extends Exception{
    private double deficit;

    public OverdraftException(String message,double deficit){
        super(message);
        this.deficit = deficit;
    }

    public double getDeficit() {
        return deficit;
    }
}
```

测试

```
package experiment.six.com.mybank.test;


import experiment.six.com.mybank.domain.*;


public class TestBanking2 {
    public static void main(String[] args) {
        Bank.addCustomer("zhang", "san");
        Customer c1 = Bank.getCustomer(0);
        c1.addAccount(new SavingsAccount(500,0.017));
        c1.addAccount(new CheckingAccount(2000, 3000));
        for(int i = 0; i < Bank.getNumbersOfCustomers(); i++)
{
            Customer cus = Bank.getCustomer(i);
            System.out.println(cus);
            for(int j = 0;j < cus.getNumbersOfAccounts();j++)
{
                Account acc = cus.getAccount(j);
                System.out.println("第"+(j+1)+"个账户");
                if(acc instanceof SavingsAccount sa){
                    System.out.printf("计算余额
%.2f\n",sa.accumulateInterest());
                }
                else if(acc instanceof CheckingAccount ck){
                    System.out.printf("当前余额
为%.2f\n",ck.getBalance());
                    System.out.printf("能够透支的钱数
%.2f\n",ck.getOverdraftAmount());
                    try {
                        ck.withdraw(6000);
                    } catch (OverdraftException e) {
                        System.out.println(e.getMessage()+"赤
字为"+e.getDeficit());
                        e.printStackTrace();
                    }
                }
                else {
                    //普通账户
```

```
                    try {
                        acc.withdraw(60000);
                    } catch (OverdraftException e) {
                        System.out.println(e.getMessage());
                        e.printStackTrace();
                    }
                    System.out.printf("余额为
%.2f",acc.getBalance());
                }
            }
        }
    }
}

```

```
C:\Users\liyaj\.jdks\openjdk-17.0.2\bin\java.exe -javaagent:C:\Users\liyaj\AppData\Local\JetBrains\Toolbox\apps\IDEA-U\ch-0\213.7172.25\lib\idea_rt
.jar=64712:C:\Users\liyaj\AppData\Local\JetBrains\Toolbox\apps\IDEA-U\ch-0\213.7172.25\bin -Dfile.encoding=UTF-8 -classpath C:\Users\liyaj\IdeaProjects\2022_Java\out\production\2022_Java
experiment.six.com.mybank.test.TestBanking2
zhang-san
第1个账户
计算余额 508.50
第2个账户
当前余额为2000.00
能够透支的钱数 3000.00
透支额度不足赤字为4000.0
experiment.six.com.mybank.domain.OverdraftException Create breakpoint : 透支额度不足
    at experiment.six.com.mybank.domain.CheckingAccount.withdraw(CheckingAccount.java:26)
    at experiment.six.com.mybank.test.TestBanking2.main(TestBanking2.java:26)

Process finished with exit code 0
```

实验七

生产者消费者

实验代码

H5 Consumer

```
package experiment.seven.text01;

/**
 * @author liyajun
 * @date 2022/4/18 14:58

```

```
 */
public class Consumer extends Thread{
    private MyStack myStack;
    public Consumer(MyStack my){
        this.myStack = my;
    }
    @Override
    public void run(){
        while(true){
            myStack.pop();
        }
    }
}
```

H5 Producer

```
package experiment.seven.text01;

/**
 * @author liyajun
 * @date 2022/4/18 14:58
 */
public class Producer extends Thread{
    private MyStack myStack;
    public Producer(MyStack my){
        this.myStack = my;
    }
    @Override
    public void run(){
        while(true){
            //if(myStack.getPoint() == MyStack.SIZE)
this.notifyAll();
            myStack.push('c');
        }
    }

}
```

H5 MyStack

```java
package experiment.seven.text01;

import java.util.List;
import java.util.Stack;

/**
 * @author liyajun
 * @date 2022/4/18 14:58
 */
public class MyStack {
    static final int SIZE = 10;
    private volatile Stack<Character> st = new Stack<>();
    public  synchronized void push(char c){
        if(st.size() > SIZE){
            System.out.println("容量已满");
            try {
                this.wait();
                this.notify();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        } else {

 System.out.println(Thread.currentThread().getName() + "生产了"+c);
            this.notify();
            st.push(c);
        }
    }
    public synchronized void pop(){
        if(st.size() == 0){
            System.out.println("容量为空请生产");
            try {
                this.wait();
                this.notify();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        } else {
```

```
38
 System.out.println(Thread.currentThread().getName() + "取走了"+ st.pop());
39              this.notify();
40          }
41      }
42      public int getPoint(){
43          return st.size();
44      }
45  }
```

H5 SyncTest

```
1   package experiment.seven.text01;
2
3   /**
4    * @author liyajun
5    * @date 2022/4/18 14:58
6    */
7   public class SyncTest {
8       public static void main(String[] args) {
9           MyStack my = new MyStack();
10          Consumer co = new Consumer(my);
11          Producer pr = new Producer(my);
12          co.start();
13          pr.start();
14      }
15  }
```

测试结果

```
Thread-0取走了c
Thread-0取走了c
容量为空请生产
Thread-1生产了c
Thread-1生产了c
Thread-1生产了c
Thread-1生产了c
Thread-1生产了c
Thread-1生产了c
Thread-1生产了c
Thread-1生产了c
Thread-1生产了c
Thread-1生产了c
容量已满
Thread-0取走了c
Thread-0取走了c
Thread-0取走了c
Thread-0取走了c
Thread-0取走了c
Thread-0取走了c
Thread-0取走了c
Thread-0取走了c
Thread-1生产了c
Thread-1生产了c
Thread-1生产了c
Thread-1生产了c
Thread-1生产了c
Thread-1生产了c
Thread-1生产了c
```

改写银行

实验代码

H5 **volatile *and* synchronized**

```java
package experiment.seven.new_bank.com.domain;

/**
 * @author liyajun
```

```
 * @date 2022/4/18 14:03
 */
public class ConcurrentAccount {
    private volatile double balance = 0.0;

    /**
     * 存钱
     * @param amt
     */
    public synchronized void deposit(double amt){
        this.notifyAll();
        balance += amt;

 System.out.println(Thread.currentThread().getName()+"存
入"+amt+"元");
    }

    /**
     * 取钱
     * @param amt
     */
    public synchronized void withdraw(double amt){
        if(balance == 0) {

 System.out.println(Thread.currentThread().getName()+"余额不
足");
            try {
                this.wait();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        if(balance >= amt){
            balance -= amt;

 System.out.println(Thread.currentThread().getName()+"取
出"+amt+"元");
        } else {

 System.out.println(Thread.currentThread().getName()+"余额不
足");
        }
    }
```

```
40
41      /**
42       * 获取余额
43       */
44      public void getBalance(){
45          System.out.println("账户余额为" + balance+"元");
46      }
47  }
48
```

H5 使用Lock

```
1   package experiment.seven.new_bank.com.domain;
2
3   import java.util.concurrent.locks.Lock;
4   import java.util.concurrent.locks.ReentrantLock;
5
6   /**
7    * @author liyajun
8    * @date 2022/4/18 14:03
9    */
10  public class ConcurrentAccount {
11      private volatile double balance = 0.0;
12      private Lock lock = new ReentrantLock();
13
14      /**
15       * 存钱
16       * @param amt
17       */
18      public void deposit(double amt){
19          lock.lock();
20          balance += amt;
21
    System.out.println(Thread.currentThread().getName()+"存
    入"+amt+"元");
22          lock.unlock();
23      }
24
25      /**
26       * 取钱
27       * @param amt
28       */
```

```
    public void withdraw(double amt){
        lock.lock();
        if(balance >= amt){
            balance -= amt;

System.out.println(Thread.currentThread().getName()+"取出"+amt+"元");
        } else {

System.out.println(Thread.currentThread().getName()+"余额不足");
        }
        lock.unlock();
    }

    /**
     * 获取余额
     */
    public void getBalance(){
        System.out.println("账户余额为" + balance+"元");
    }
}
```

H5 test

```
package experiment.seven.new_bank.com.test;

import experiment.seven.new_bank.com.domain.ConcurrentAccount;

/**
 * @author liyajun
 * @date 2022/4/18 14:09
 */
public class TestBanking3 {
    public static void main(String[] args) {
        ConcurrentAccount ca = new ConcurrentAccount();
        new Thread(() -> {
            while (true){
                ca.deposit(114);
                try {
                    Thread.sleep(300);
                } catch (InterruptedException e) {

```

```
                    e.printStackTrace();
                }
                ca.getBalance();
            }
        }).start();
        new Thread(() -> {
            while (true){
                ca.withdraw(114);
                try {
                    Thread.sleep(300);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
                ca.getBalance();
            }
        }).start();
    }
}
```

运行结果

```
Thread-0存入114.0元
Thread-1取出114.0元
账户余额为0.0元
Thread-0存入114.0元
账户余额为114.0元
Thread-1取出114.0元
账户余额为0.0元
账户余额为0.0元
Thread-1余额不足
Thread-0存入114.0元
Thread-1取出114.0元
账户余额为0.0元
账户余额为0.0元
Thread-1余额不足
Thread-0存入114.0元
Thread-1取出114.0元
账户余额为0.0元
Thread-0存入114.0元
账户余额为0.0元
Thread-1取出114.0元
账户余额为0.0元
账户余额为0.0元
Thread-0存入114.0元
Thread-1取出114.0元
账户余额为0.0元
账户余额为0.0元
Thread-0存入114.0元
```

改进版猜数字

代码

GenNum

生成数字

```java
package experiment.seven.text03;

import java.util.Random;

/**
 * @author liyajun
 * @date 2022/4/18 16:02
 */
```

```java
public class GenNum implements Runnable{
    int num;
    @Override
    public synchronized void run(){
        Random ran = new Random();
        this.num = ran.nextInt(100);
    }
    public int getNum(){
        return num;
    }
    public void  setNum(int num){
        this.num = num;
    }
}
```

H5 猜数字

```java
package experiment.seven.text03;

/**
 * @author liyajun
 * @date 2022/4/18 15:56
 * 出数字的线程为守护线程
 */
public class Num {

    public static void main(String[] args) {
        GenNum n1 = new GenNum();
        Thread t1 = new Thread(n1);
        t1.run();
        t1.setDaemon(true);
        try {
            Thread.sleep(100);
            GenNum n2 = new GenNum();
            Thread t2 = new Thread(n2);
            while(true){
                t2.interrupt();
                t2.run();
                Thread.sleep(100);
                t1.interrupt();
```

```
                if(n1.getNum() == n2.getNum()){
                    System.out.println("猜的数字为"+n2.getNum());
                    System.out.println("猜对了!");
                    break;
                } else if(n1.getNum() > n2.getNum()){
                    System.out.println("猜的数字为"+n2.getNum());
                    System.out.println("猜小了");
                } else {
                    System.out.println("猜的数字为"+n2.getNum());
                    System.out.println("猜大了");
                }
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

    }
}
```

测试结果

```
C:\Users\liyaj\.jdks\openjdk-17.0.2\bin\java.exe -javaagent:C:\Users\liyaj\AppData\Local\JetBrains\Toolbox\apps\IDEA-U\ch-0\213.7172.25\lib\idea_rt
.jar=52053:C:\Users\liyaj\AppData\Local\JetBrains\Toolbox\apps\IDEA-U\ch-0\213.7172.25\bin -Dfile.encoding=UTF-8 -classpath C:\Users\liyaj\IdeaProjects\2022_Java\out\pro
experiment.seven.text03.Num
猜的数字为90
猜大了
猜的数字为73
猜小了
猜的数字为14
猜小了
猜的数字为39
猜小了
猜的数字为88
猜大了
猜的数字为95
猜大了
猜的数字为95
猜大了
猜的数字为69
猜小了
猜的数字为16
猜小了
猜的数字为76
猜小了
猜的数字为3
猜小了
猜的数字为81
猜大了
猜的数字为57
猜小了
猜的数字为85
猜大了
猜的数字为41
猜小了
猜的数字为80
猜对了!

Process finished with exit code 0
```

尾

单纯的随机猜靠运气，没有给另一个线程加上二分，可以加上二分提速。