Exercise 05

班级：2017211106 学号：2017212116 姓名：杨诺诚

问题一：ATM 游戏机

解决思路：

创建一个以 Account 为类型的数组 player，可以调用每一个元素的属性值，分别操作互不影响。

代码实现：

调用类：

```
package com.company;

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

public class Main {

    public static void main(String[] args) {
//        List<Account> players = new ArrayList<Account>();
        Scanner input = new Scanner(System.in);
        int id = 0;
        int choice = 0;
        Account players[];
        players = new Account[10];
        for(int i=0;i<players.length;i++) {
            players[i] = new Account(i+1,100);
        }
        while (true){
            while (true) {
            System.out.print("Enter an id: ");
            id = input.nextInt();
            if (id < 0 || id > 10)
                System.out.println("Please try again!");
            else break;
            }
            while(true) {
                System.out.print("\nMain menu:\n1: Check balance\n2: withdraw\n3:
deposit\n4: exit");
```

```
                System.out.print("\nEnter a choice:");
                choice = input.nextInt();
                if (choice == 1) {
                    System.out.println("The balabce is " +
players[id].getBalance());
                } else if (choice == 2) {
                    System.out.print("Enter an amount to withdraw: ");
                    players[id].withDraw(input.nextDouble());
                } else if (choice == 3) {
                    System.out.print("Enter an amount to deposit: ");
                    players[id].deposit(input.nextDouble());
                } else {
                    break;
                }
            }
        }
    }
}
```

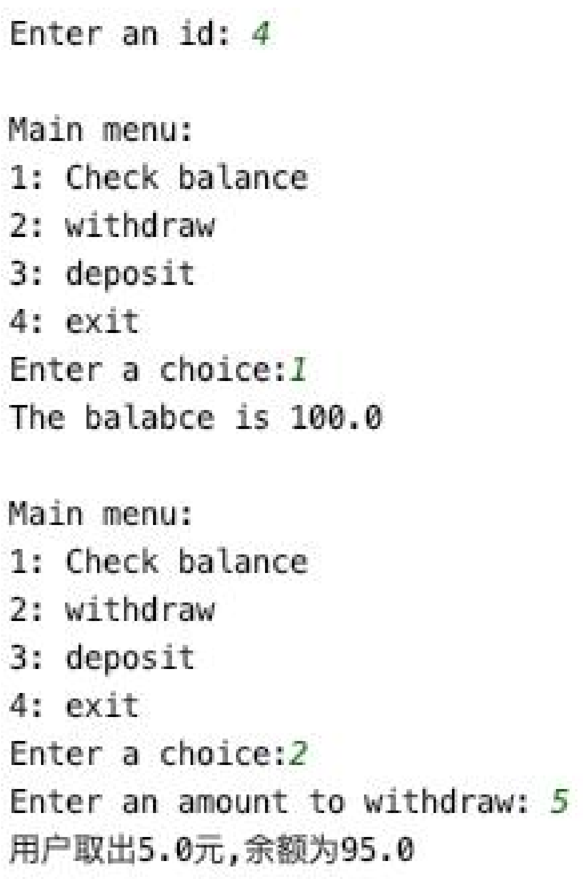
Account 类:

```
package com.company;

import java.util.ArrayList;
import java.util.Date;

class Account{
    private int id = 0;
    private double balance = 0.0;//本金
    private double annualInterestRate = 0.0;//年利率
    private Date dateCreated;
    private ArrayList<Double> history_Value = new ArrayList<Double>();
    private ArrayList<Date> history_Date = new ArrayList<Date>();
    Account(){
        dateCreated = new Date();
//            无参构造方法
    }
    Account(int id ,double balance){
//        System.out.println("this.id = "+ this.id+'\n'+"id = "+id);
        dateCreated = new Date();
        this.id = id;
        this.balance = balance;
    }
    int getId(){
        return this.id;
    }
    void setId(int id){
        this.id = id;
    }
    String getBalance(){
        return String.valueOf(this.balance);
    }
    void setBalance(double balance){
        this.balance = balance;
    }
    double getAnnualInterestRate(){
        return this.annualInterestRate;
    }
    void setannualInterestRate(double annualInterestRate){
        this.annualInterestRate = annualInterestRate/100;
//            由小数转化为百分数（ 75 -> 75% ）
    }
    Date getdateCreate(){
        return this.dateCreated;
    }
    double getMonthlyInterestRate(){
        return balance*(annualInterestRate/12.0);
    }
    void withDraw(double balance){
        if(balance<=this.balance){
            this.balance -= balance;
            System.out.println("用户取出"+balance+"元,"+"余额为"+this.balance);
            history_Value.add(-1*balance);
            history_Date.add(new Date());
        }
        else System.out.println("余额不足，请重新输入！");
    }
    void deposit(double balance){
        this.balance += balance;
        System.out.println("用户存入"+balance+"元,"+"余额为"+this.balance);
        history_Value.add(balance);
        history_Date.add(new Date());
    }
    void showHistory(){
        for(int i=0;i<history_Value.size();i++){
            System.out.println(history_Date.get(i).toString()+" & 
"+history_Value.get(i).toString());
        }
        for(int i=0;i<history_Date.size();i++){
            if(i!=0) System.out.println("");
            System.out.print("用户在 "+history_Date.get(i).toString());
//              System.out.print((history_Value.get(i)<0 ? " 取出 ":" 存入 ") +
Math.abs(history_Value.get(i))+" 元");
            System.out.print(history_Value.get(i)<0 ? (" 取出 "+ -1 *
history_Value.get(i)+" 元"):(" 存入 " + history_Value.get(i)+" 元"));
        }
    }
}
```

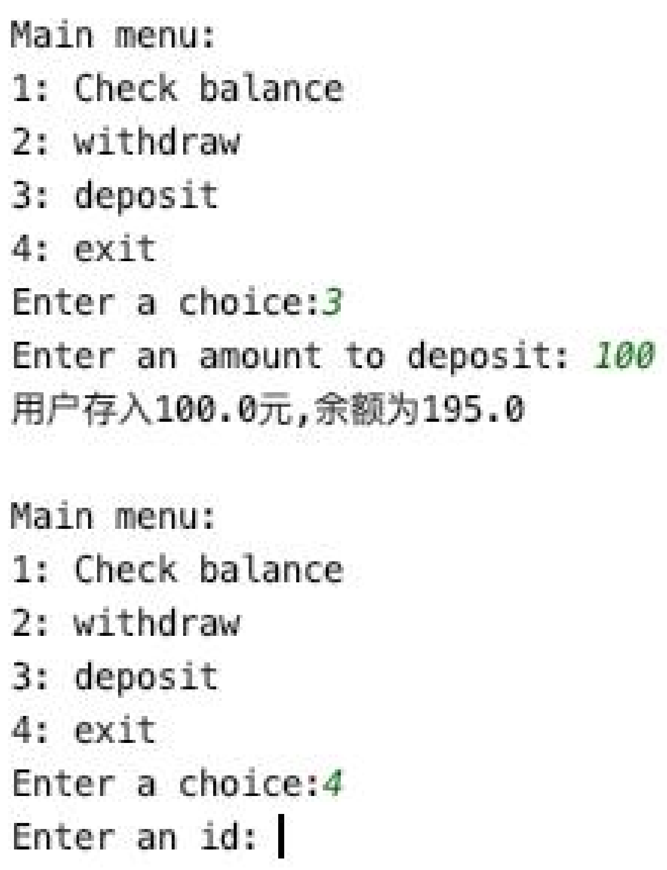
效果图展示:



```
Enter an id: 4

Main menu:
1: Check balance
2: withdraw
3: deposit
4: exit
Enter a choice:1
The balabce is 100.0

Main menu:
1: Check balance
2: withdraw
3: deposit
4: exit
Enter a choice:2
Enter an amount to withdraw: 5
用户取出5.0元,余额为95.0
```



```
Main menu:
1: Check balance
2: withdraw
3: deposit
4: exit
Enter a choice:3
Enter an amount to deposit: 100
用户存入100.0元,余额为195.0

Main menu:
1: Check balance
2: withdraw
3: deposit
4: exit
Enter a choice:4
Enter an id: 
```

问题二：字符串 split 方法

解决思路：

本类中主要要实现字符串的匹配于位置记录的功能。

有两个方法可以实现：

1）重载 **String** 方法中的 **split** 类

2）定义 **Newsplit** 类中的 **split** 方法

选择方法二实现。定义新的方法 **split**。

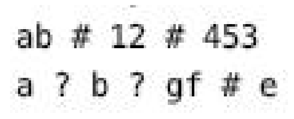
在这个方法中，先将字符串中的所有内容放置在先进先出的队列（**Queue**）中，将待监测的字符放置在集合（**Hashset**）以中达到更高的运算速度，每次将队列中的下一个字符与集合内容相比较，若相同，则把它上一次匹配之后出队的所有元素添加到列表（**ArrayList**）中，若不匹配则将它存储到一个字符串中（**Changeable**）。

代码实现：

```
package com.company;
import java.lang.String;
import java.util.*;

public class NewSplit {
   public static String[] split(String s , String regex){
      Queue<Character> queue = new LinkedList<Character>();
      ArrayList<String> arrayList = new ArrayList<String>();
      Set<Character> set = new HashSet<Character>();
      for(int i = 0;i < s.length(); i++){
         //将 s 全部录入队列中
         queue.add(s.charAt(i));
      }
      for (int i = 0;i < regex.length();i++){
         //将待匹配字符串录入 Hashset 中
         if(regex.charAt(i)!='['||regex.charAt(i)!=']'){
            set.add(regex.charAt(i));
         }
      }
      String Changeable = "";
      //拼接一个完整的字符串
      while(true){
         char CharAtFirst = queue.poll();
         if(set.contains(CharAtFirst)){
            arrayList.add(Changeable);
            arrayList.add(String.valueOf(CharAtFirst));
            Changeable = "";
         }
         else{
                Changeable += CharAtFirst;
            }
            if(queue.isEmpty()){
                if(!Changeable.equals(""))
                    arrayList.add(Changeable);
                break;//没有元素为止
            }
        }
        String [] ans = new String[arrayList.size()];
        for(int i = 0;i < ans.length;i++){
            ans[i] = arrayList.get(i);
        }
        return ans;
    }
    public static void main(String[] args) {
        String s = "ab#12#453";
        String regex = "#";
        String [] Newsplit = split(s,regex);
        for(int i = 0;i < Newsplit.length;i++){
            System.out.print(" "+Newsplit[i]);
        }
        System.out.println();
        s = "a?b?gf#e";
        regex = "[?#]";
        String [] Newsplit2 = split(s,regex);
        for(int i = 0;i < Newsplit2.length;i++){
            System.out.print(" "+Newsplit2[i]);
        }
    }
}
```

效果图展示:



```
ab # 12 # 453
a ? b ? gf # e
```

在此例子中，使用了较多的存储方法如 **list**，**set**，**queue** 达到了很高的灵活度，极大的提高了程序编写和阅读的方便性。