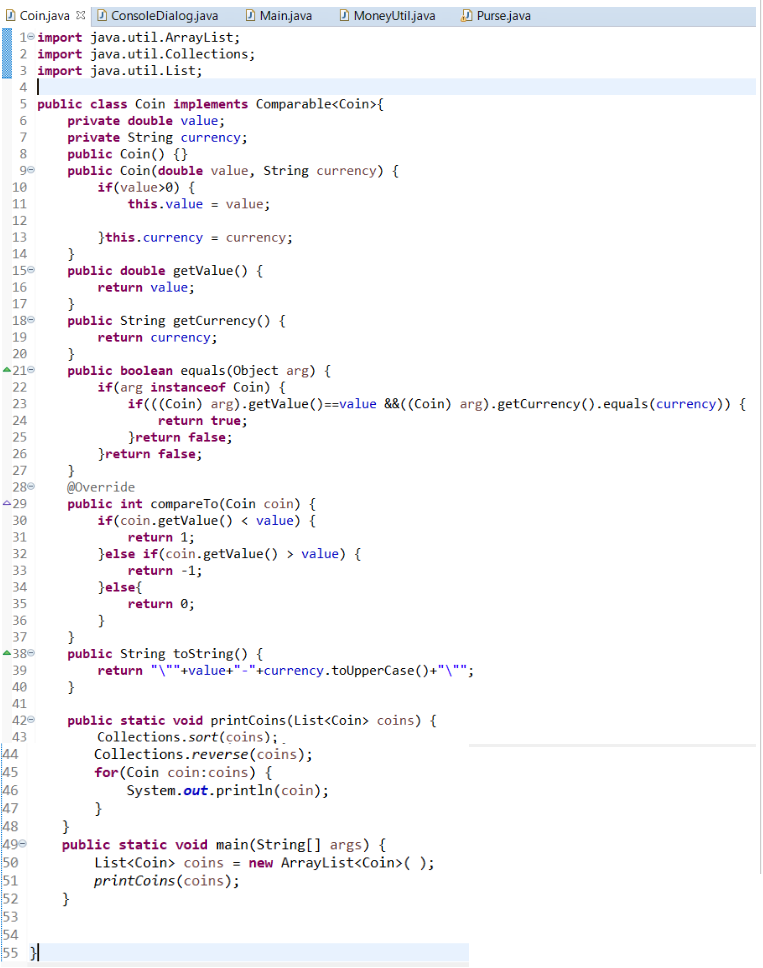
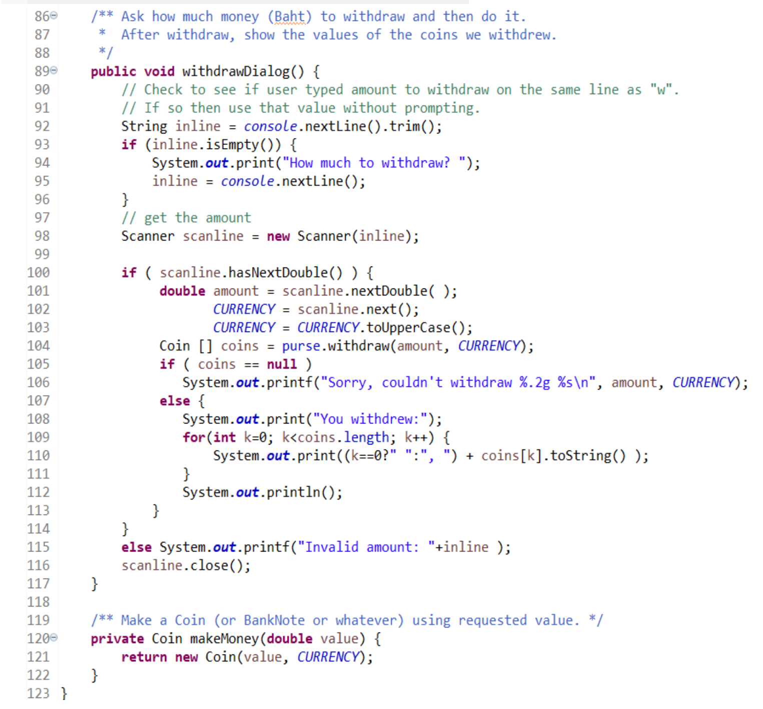
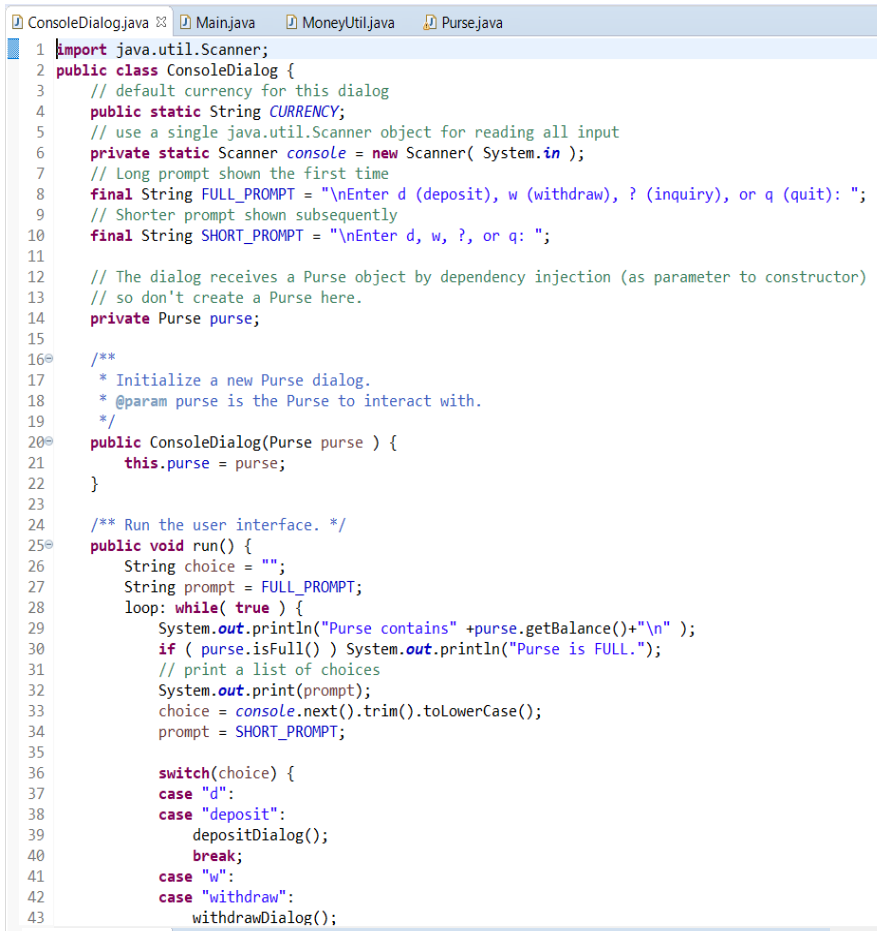
Lab Coin



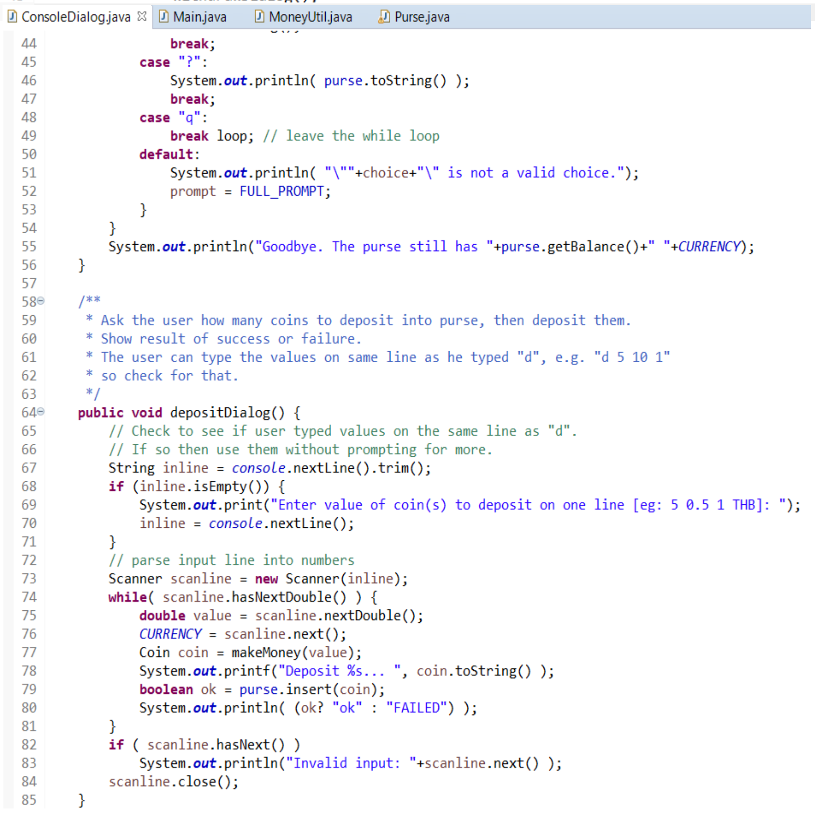
```java
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class Coin implements Comparable<Coin>{
    private double value;
    private String currency;
    public Coin() {}
    public Coin(double value, String currency) {
        if(value>0) {
            this.value = value;

        }this.currency = currency;
    }
    public double getValue() {
        return value;
    }
    public String getCurrency() {
        return currency;
    }
    public boolean equals(Object arg) {
        if(arg instanceof Coin) {
            if(((Coin) arg).getValue()==value &&((Coin) arg).getCurrency().equals(currency)) {
                return true;
            }return false;
        }return false;
    }
    @Override
    public int compareTo(Coin coin) {
        if(coin.getValue() < value) {
            return 1;
        }else if(coin.getValue() > value) {
            return -1;
        }else{
            return 0;
        }
    }
    public String toString() {
        return "\""+value+"-"+currency.toUpperCase()+"\"";
    }

    public static void printCoins(List<Coin> coins) {
        Collections.sort(coins);
       Collections.reverse(coins);
       for(Coin coin:coins) {
           System.out.println(coin);
       }
    }
   public static void main(String[] args) {
       List<Coin> coins = new ArrayList<Coin>( );
       printCoins(coins);
   }


}
```



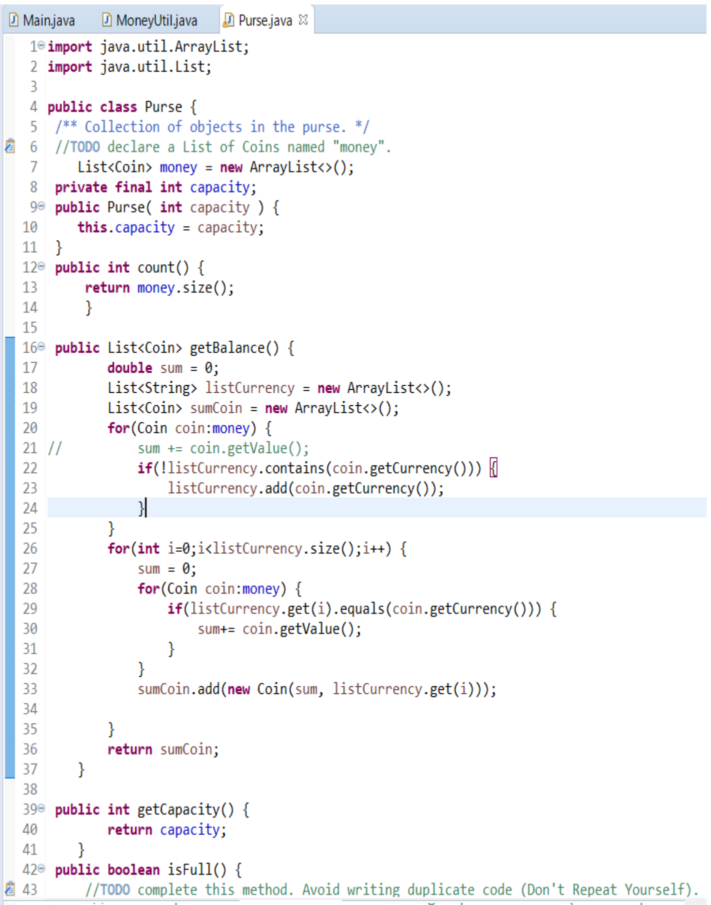
```java
    /** Ask how much money (Baht) to withdraw and then do it.
     *  After withdraw, show the values of the coins we withdrew.
     */
    public void withdrawDialog() {
        // Check to see if user typed amount to withdraw on the same line as "w".
        // If so then use that value without prompting.
        String inline = console.nextLine().trim();
        if (inline.isEmpty()) {
            System.out.print("How much to withdraw? ");
            inline = console.nextLine();
        }
        // get the amount
        Scanner scanline = new Scanner(inline);

        if ( scanline.hasNextDouble() ) {
            double amount = scanline.nextDouble( );
                CURRENCY = scanline.next();
                CURRENCY = CURRENCY.toUpperCase();
            Coin [] coins = purse.withdraw(amount, CURRENCY);
            if ( coins == null )
                System.out.printf("Sorry, couldn't withdraw %.2g %s\n", amount, CURRENCY);
            else {
                System.out.print("You withdrew:");
                for(int k=0; k<coins.length; k++) {
                    System.out.print((k==0?" ":", ") + coins[k].toString() );
                }
                System.out.println();
            }
        }
        else System.out.printf("Invalid amount: "+inline );
        scanline.close();
    }

    /** Make a Coin (or BankNote or whatever) using requested value. */
    private Coin makeMoney(double value) {
        return new Coin(value, CURRENCY);
    }
}
```



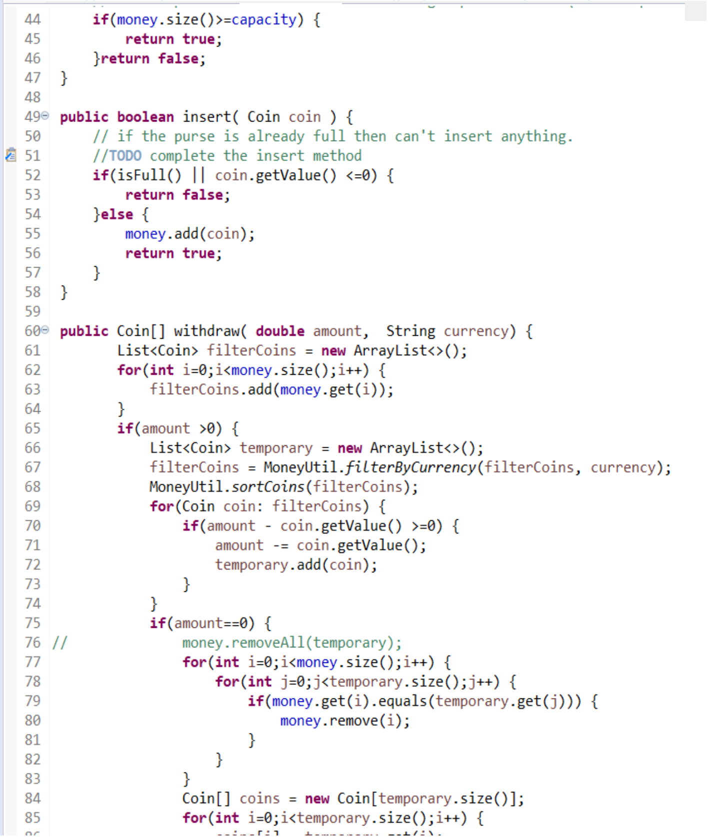
```java
import java.util.Scanner;
public class ConsoleDialog {
    // default currency for this dialog
    public static String CURRENCY;
    // use a single java.util.Scanner object for reading all input
    private static Scanner console = new Scanner( System.in );
    // Long prompt shown the first time
    final String FULL_PROMPT = "\nEnter d (deposit), w (withdraw), ? (inquiry), or q (quit): ";
    // Shorter prompt shown subsequently
    final String SHORT_PROMPT = "\nEnter d, w, ?, or q: ";

    // The dialog receives a Purse object by dependency injection (as parameter to constructor)
    // so don't create a Purse here.
    private Purse purse;

    /**
     * Initialize a new Purse dialog.
     * @param purse is the Purse to interact with.
     */
    public ConsoleDialog(Purse purse ) {
        this.purse = purse;
    }

    /** Run the user interface. */
    public void run() {
        String choice = "";
        String prompt = FULL_PROMPT;
        loop: while( true ) {
            System.out.println("Purse contains" +purse.getBalance()+"\n" );
            if ( purse.isFull() ) System.out.println("Purse is FULL.");
            // print a list of choices
            System.out.print(prompt);
            choice = console.next().trim().toLowerCase();
            prompt = SHORT_PROMPT;

            switch(choice) {
            case "d":
            case "deposit":
                depositDialog();
                break;
            case "w":
            case "withdraw":
                withdrawDialog();
```



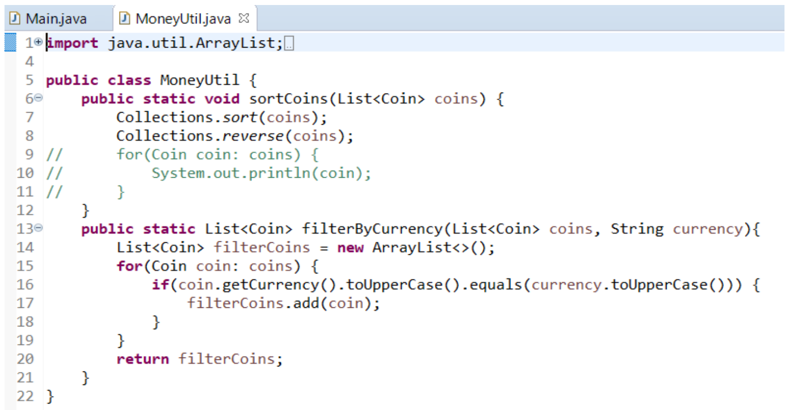
```java
                break;
            case "?":
                System.out.println( purse.toString() );
                break;
            case "q":
                break loop; // leave the while loop
            default:
                System.out.println( "\""+choice+"\" is not a valid choice.");
                prompt = FULL_PROMPT;
            }
        }
        System.out.println("Goodbye. The purse still has "+purse.getBalance()+" "+CURRENCY);
    }

    /**
     * Ask the user how many coins to deposit into purse, then deposit them.
     * Show result of success or failure.
     * The user can type the values on same line as he typed "d", e.g. "d 5 10 1"
     * so check for that.
     */
    public void depositDialog() {
        // Check to see if user typed values on the same line as "d".
        // If so then use them without prompting for more.
        String inline = console.nextLine().trim();
        if (inline.isEmpty()) {
            System.out.print("Enter value of coin(s) to deposit on one line [eg: 5 0.5 1 THB]: ");
            inline = console.nextLine();
        }
        // parse input line into numbers
        Scanner scanline = new Scanner(inline);
        while( scanline.hasNextDouble() ) {
            double value = scanline.nextDouble();
            CURRENCY = scanline.next();
            Coin coin = makeMoney(value);
            System.out.printf("Deposit %s... ", coin.toString() );
            boolean ok = purse.insert(coin);
            System.out.println( (ok? "ok" : "FAILED") );
        }
        if ( scanline.hasNext() )
            System.out.println("Invalid input: "+scanline.next() );
        scanline.close();
    }
```



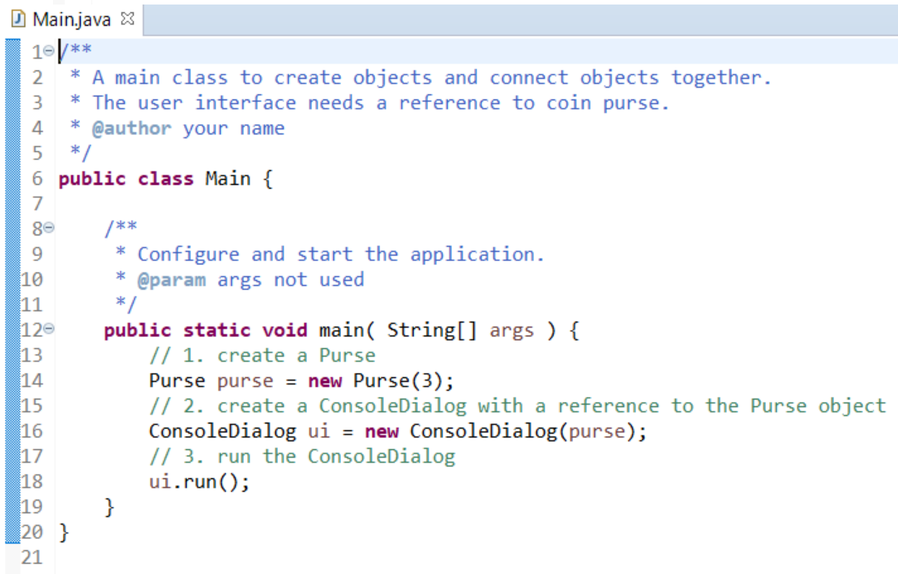
```java
import java.util.ArrayList;
import java.util.List;

public class Purse {
 /** Collection of objects in the purse. */
 //TODO declare a List of Coins named "money".
    List<Coin> money = new ArrayList<>();
 private final int capacity;
 public Purse( int capacity ) {
     this.capacity = capacity;
 }
 public int count() {
     return money.size();
     }

 public List<Coin> getBalance() {
        double sum = 0;
        List<String> listCurrency = new ArrayList<>();
        List<Coin> sumCoin = new ArrayList<>();
        for(Coin coin:money) {
//          sum += coin.getValue();
            if(!listCurrency.contains(coin.getCurrency())) {
                listCurrency.add(coin.getCurrency());
            }
        }
        for(int i=0;i<listCurrency.size();i++) {
            sum = 0;
            for(Coin coin:money) {
                if(listCurrency.get(i).equals(coin.getCurrency())) {
                    sum+= coin.getValue();
                }
            }
            sumCoin.add(new Coin(sum, listCurrency.get(i)));

        }
        return sumCoin;
    }

 public int getCapacity() {
        return capacity;
    }
 public boolean isFull() {
    //TODO complete this method. Avoid writing duplicate code (Don't Repeat Yourself).
```



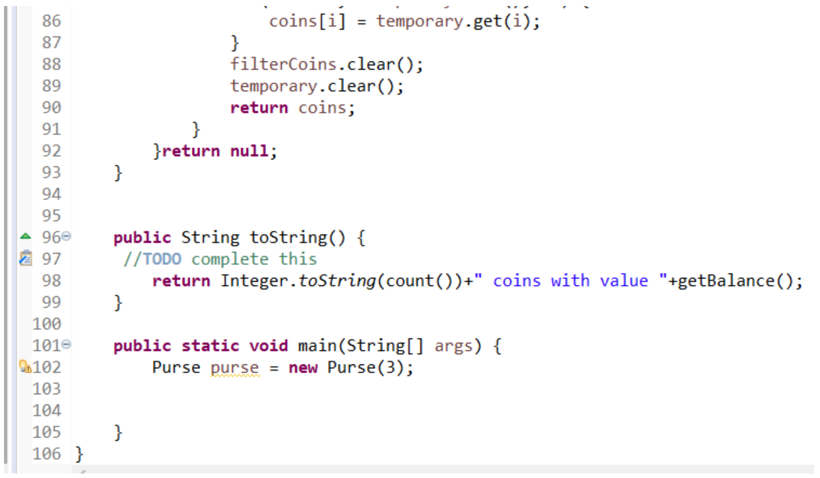
```java
     if(money.size()>=capacity) {
         return true;
     }return false;
 }

 public boolean insert( Coin coin ) {
     // if the purse is already full then can't insert anything.
     //TODO complete the insert method
     if(isFull() || coin.getValue() <=0) {
         return false;
     }else {
         money.add(coin);
         return true;
     }
 }

 public Coin[] withdraw( double amount,  String currency) {
        List<Coin> filterCoins = new ArrayList<>();
        for(int i=0;i<money.size();i++) {
            filterCoins.add(money.get(i));
        }
        if(amount >0) {
            List<Coin> temporary = new ArrayList<>();
            filterCoins = MoneyUtil.filterByCurrency(filterCoins, currency);
            MoneyUtil.sortCoins(filterCoins);
            for(Coin coin: filterCoins) {
                if(amount - coin.getValue() >=0) {
                    amount -= coin.getValue();
                    temporary.add(coin);
                }
            }
            if(amount==0) {
//              money.removeAll(temporary);
                for(int i=0;i<money.size();i++) {
                    for(int j=0;j<temporary.size();j++) {
                        if(money.get(i).equals(temporary.get(j))) {
                            money.remove(i);
                        }
                    }
                }
                Coin[] coins = new Coin[temporary.size()];
                for(int i=0;i<temporary.size();i++) {
```



```java
import java.util.ArrayList;

public class MoneyUtil {
    public static void sortCoins(List<Coin> coins) {
        Collections.sort(coins);
        Collections.reverse(coins);
//      for(Coin coin: coins) {
//          System.out.println(coin);
//      }
    }
    public static List<Coin> filterByCurrency(List<Coin> coins, String currency){
        List<Coin> filterCoins = new ArrayList<>();
        for(Coin coin: coins) {
            if(coin.getCurrency().toUpperCase().equals(currency.toUpperCase())) {
                filterCoins.add(coin);
            }
        }
        return filterCoins;
    }
}
```



```java
/**
 * A main class to create objects and connect objects together.
 * The user interface needs a reference to coin purse.
 * @author your name
 */
public class Main {

    /**
     * Configure and start the application.
     * @param args not used
     */
    public static void main( String[] args ) {
        // 1. create a Purse
        Purse purse = new Purse(3);
        // 2. create a ConsoleDialog with a reference to the Purse object
        ConsoleDialog ui = new ConsoleDialog(purse);
        // 3. run the ConsoleDialog
        ui.run();
    }
}
```



```java
                    coins[i] = temporary.get(i);
                }
                filterCoins.clear();
                temporary.clear();
                return coins;
            }
        }return null;
    }


    public String toString() {
     //TODO complete this
        return Integer.toString(count())+" coins with value "+getBalance();
    }

    public static void main(String[] args) {
        Purse purse = new Purse(3);


    }
}
```