class Sequence{

    static Sequence obj=null;

    static int count;

    private Sequence(){

        count=0;

    }

    static synchronized Sequence getInstance(){

        if(obj==null){

            obj=new Sequence();

            System.out.println("Objcet created");

        }

        count++;

        System.out.println("Count is "+count);

        return obj;

    }

}

public class Demo {

    public static void main(String[] args) {

        Sequence s1=Sequence.getInstance();

        Sequence s2=Sequence.getInstance();

        Sequence s3=Sequence.getInstance();

        Sequence s4=Sequence.getInstance();

    }

}

abstract class Account{

    abstract void calculateInterest();

}

class Saving extends Account{

    @Override

    void calculateInterest() {

       System.out.println("Calculate inerest of saving");

    }

}

class Loan extends Account{

    @Override

    void calculateInterest() {

       System.out.println("Calculate inerest of Loan");

    }

}

class AccountFactory{

    static Account getAccount(String type){

        Account obj=null;

        if(type.equalsIgnoreCase("saving")){

            obj=new Saving();

        }else if (type.equalsIgnoreCase("loan")){

            obj=new Loan();

        }

        return obj;

    }

}

public class Demo2 {

public static void main(String[] args) {

    Account ob1=AccountFactory.getAccount("loan");

    Account ob2=AccountFactory.getAccount("saving");

    ob1.calculateInterest();

    ob2.calculateInterest();

}

}

import java.util.ArrayList;

class Student implements Cloneable {

    ArrayList<String> ar;

    Student() {

        ar = new ArrayList<>();

    }

    Student(ArrayList<String> t) {

        ar = t;

    }

    void loadData() {

        ar.add("A");

        ar.add("B");

        ar.add("C");

        ar.add("D");

    }

    void show() {

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

        for (String s : ar) {

            System.out.println(s);

        }

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

    }

    @Override

    protected Object clone() throws CloneNotSupportedException {

        ArrayList<String> ar2;

        ar2 = new ArrayList<>();

        for (String s : ar) {

            ar2.add(s);

        }

        Student obj = new Student(ar2);

        return obj;

    }

}

public class Demo3 {

    public static void main(String[] args)throws Exception

     {

        Student st1=new Student();

        st1.loadData();

        st1.show();

         Student st2=(Student)st1.clone();

       // Student st2=st1;

        st2.show();

        st2.ar.remove("C");

        System.out.println("\*\*st1 object\*\*");

        st1.show();

        System.out.println("\*\*st2 object\*\*");

        st2.show();

    }

}