```
1.  import java.util.*;
2.
3.  class Repo {
4.      private static ArrayList<String> missions = new ArrayList<String>();
5.      private static boolean k = true;
6.
7.      public Repo(String items) {
8.          String[] itemss = items.split("\\s+");
9.          for(String item : itemss)
10.             missions.add(item);
11.     }
12.
13.     int getSize() {
14.         return missions.size();
15.     }
16.
17.     public synchronized void give1() {
18.         if(k == false) {
19.             try {
20.                 this.wait();
21.             }
22.             catch(Exception e) {
23.                 e.printStackTrace();
24.             }
25.         }
26.         else {
27.             System.out.println(Thread.currentThread().getName() + " finish "
    + missions.get(0));
28.             missions.remove(0);
29.             k = false;
30.             this.notify();
31.         }
32.     }
33.
34.     public synchronized void give2() {
35.         if(k == true) {
36.             try {
37.                 this.wait();
38.             }
39.             catch(Exception e) {
40.                 e.printStackTrace();
41.             }
42.         }
43.         else {
```

```
44.             System.out.println(Thread.currentThread().getName()+ " finish "
    + missions.get(0));
45.             missions.remove(0);
46.             k = true;
47.             this.notify();
48.         }
49.     }
50. }
51.
52. class Worker1 implements Runnable {
53.     Repo repo;
54.     Worker1(Repo repo) {
55.         this.repo = repo;
56.     }
57.
58.     public void run() {
59.         while(true) {
60.             if(repo.getSize() == 0) break;
61.             repo.give1();
62.         }
63.     }
64. }
65.
66. class Worker2 implements Runnable {
67.     Repo repo;
68.     Worker2(Repo repo) {
69.         this.repo = repo;
70.     }
71.
72.     public void run() {
73.         while(true) {
74.             if(repo.getSize() == 0) break;
75.             repo.give2();
76.         }
77.     }
78. }
```

```java
import java.util.*;

class Student{
    private String name;
    private String sex;
    private int age;

    @Override
    public String toString() {
        String str="Student [name='"+name+"', sex='"+sex+"', age="+age+"]";

        return str;
    }

    public Student(String name, String sex, int age) {
        this.name = name;
        this.sex = sex;
        this.age = age;
    }

    public String getName() {
        return name;
    }

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

    public String getSex() {
        return sex;
    }

    public void setSex(String sex) {
        this.sex = sex;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        String name=sc.next();
        int age=Integer.valueOf(sc.next());
        String sx=sc.next();
        Student student = new Student(name, sx, age);
        System.out.println(student);
    }
}
```

```java
import java.util.*;
public class Main {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        int []a = new int [5];
        while(true)
        {
            String first;
            first = sc.next();
            if(first.equals("other"))
                break;
            if(first.equals("arr"))
            {
                try {
                    int secend = sc.nextInt();
                    int t = a[secend];
                }catch(Exception e){
                    System.out.println(e);
                }
            }
            if(first.equals("null"))
            {
                try {
                    String t = null;
                    int length = t.length();
                }catch(Exception e) {
                    System.out.println(e);
                }
            }
            if(first.equals("cast"))
            {
                try {
                    Object ss = new String("string");
                    //Integer t = (Integer)ss;
                    System.out.println((Integer)ss);
                }catch(Exception e){
                    System.out.println(e);
                }
            }
            if(first.equals("num"))
            {
                try {
```

```
                    String c = sc.next();
                    Integer num = Integer.parseInt(c);
                    //System.out.println(Integer.parseInt(c));
                }catch(Exception e)
                {
                    System.out.println(e);
                }
            }

        }
        sc.close();
    }
}
```

```java
import java.util.*;

class IllegalScoreException extends Exception{
    public IllegalScoreException() {
    }

    public IllegalScoreException(String message) {
        super(message);
    }

    public IllegalScoreException(String message, Throwable cause) {
        super(message, cause);
    }

    public IllegalScoreException(Throwable cause) {
        super(cause);
    }
}

class IllegalNameException extends Exception{
    public IllegalNameException() {
    }

    public IllegalNameException(String message) {
        super(message);
    }

    public IllegalNameException(String message, Throwable cause) {
        super(message, cause);
    }

    public IllegalNameException(Throwable cause) {
        super(cause);
    }
}

class Student{
    private String name;
    private int score;
    //一般情况下，需要注意若程序抛出一个异常后，程序停止执行
    private int flag=0;
    public String getName() {
        return name;
    }
```

```
45.
46.     public int getFlag() {
47.         return flag;
48.     }
49.
50.     public void setFlag(int flag) {
51.         this.flag = flag;
52.     }
53.
54.     public void setName(String name) {
55.         char c=name.charAt(0);
56.         if (c>='0'&&c<='9'){
57.             try {
58.                 flag=1;
59.                 throw new IllegalNameException();
60.             } catch (IllegalNameException e) {
61.                 System.out.println("IllegalNameException: the first char of
    name must not be digit, name="+name);
62.             }
63.
64.             return;
65.         }
66.         this.name = name;
67.     }
68.
69.     public int getScore() {
70.         return score;
71.     }
72.
73.     public void setScore(int score)  {
74.         if (flag==1){
75.             return;
76.         }
77.         if (score<0||score>100){
78.             try {
79.                 flag=1;
80.                 throw new IllegalScoreException();
81.             } catch (IllegalScoreException e) {
82.                 System.out.println("IllegalScoreException: score out of rang
    e, score="+score);
83.             }
84.         }
85.         this.score = score;
86.     }
```

```
    //如果加分后分数<0 或>100，则抛出 IllegalScoreException，加分不成功。
    public int addScore(int score) {
        return 0;
    }

    @Override
    public String toString() {
        return "Student [" +
                "name=" + name  +
                ", score=" + score +
                ']';
    }
}

public class Main {
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        while (true){
            String str=sc.nextLine();
            if (str.equals("new")){
                String s=sc.nextLine();
                String[] s1 = s.split(" ");
                if (s1.length==2){
                    String name=s1[0];
                    int score=Integer.valueOf(s1[1]);
                    Student student = new Student();
                    student.setName(name);
                    student.setScore(score);
                    if (student.getFlag()==0){
                        System.out.println(student);
                    }
                }else{
                    System.out.println("java.util.NoSuchElementException");

                }

            }else {
                break;
            }
        }
        sc.close();
        System.out.println("scanner closed");
    }
}
```

```
class WorkerList{
    Worker readInWorker() {
        Scanner sc=new Scanner(System.in);
        Worker a= new Worker();
        String nam=sc.next();
        double sal=sc.nextDouble();
        a.setSalary(sal);
        a.setName(nam);
        sc.close();
        return a;
    }
    List<Worker> constructWorkerList()
    {
        Scanner sc=new Scanner(System.in);
        List<Worker> list=new ArrayList<Worker>();
        int n=sc.nextInt();
        //Worker w=new Worker();

        String nc;
        double sa;
        for(int i=0;i<n;i++)
        {
            Worker w=new Worker();
            //w=readInWorker();
            nc=sc.next();
            sa=sc.nextDouble();
            w.setName(nc);
            w.setSalary(sa);
            list.add(w);

        }
        sc.close();
        return list;
    }
    double computeTotalSalary(List<Worker> list)
    {
        double sum=0;
        for(int i=0;i<list.size();i++)
        {
            sum=sum+list.get(i).getSalary();
        }
        return sum;
    }
}
```

```
174. public static List<String> convertStringToList(String line)
175.     {
176.         String[] ss = line.split("\\s+");//String 的 split 方法支持正则表达
    式
177.         List<String> list = new ArrayList<String>(Arrays.asList(ss));//将数
    组转化为 list
178.         return list;
179.     }
180.     public static void remove(List<String> list, String str)
181.     {
182.             if(!list.contains(str))
183.                 return;
184.             else
185.             {
186.                 for(int i = 0;i < list.size();i++)
187.                 {
188.                     if(str.contains(list.get(i)))
189.                     {
190.                         list.remove(list.get(i));
191.                         i--;//这是难度，删除的话必须加这句，否则你可能会出现跳
    着删除的情况，如样例一
192.                     }
193.
194.                 }
195.             }
196.     }
```

```java
import java.util.ArrayList;
import java.util.Scanner;

interface GeneralStack<T>{
    public T push(T item);            //如 item 为 null，则不入栈直接返回 null。
    public T pop();                   //出栈，如为栈为空，则返回 null。
    public T peek();                  //获得栈顶元素，如为空，则返回 null.
    public boolean empty();//如为空返回 true
    public int size();     //返回栈中元素数量
}
class ArrayListGeneralStack implements GeneralStack{
    ArrayList list=new ArrayList();
    @Override
    public String toString() {
        return  list.toString();
    }

    @Override
    public Object push(Object item) {
        if (list.add(item)){
            return item;
        }else {
            return false;
        }
    }

    @Override
    public Object pop() {
        if (list.size()==0){
            return null;
        }
        return list.remove(list.size()-1);
    }

    @Override
    public Object peek() {
        return list.get(list.size()-1);
    }

    @Override
    public boolean empty() {
        if (list.size()==0){
            return true;
```

```
        }else {
            return false;
        }
    }

    @Override
    public int size() {
        return list.size();
    }
}
class Car{
    private int id;
    private String name;

    @Override
    public String toString() {
        return "Car [" +
                "id=" + id +
                ", name=" + name  +
                ']';
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

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

    public Car(int id, String name) {
        this.id = id;
        this.name = name;
    }
}
public class Main {
```

```
88.     public static void main(String[] args) {
89.         Scanner sc=new Scanner(System.in);
90.         while (true){
91.             String s=sc.nextLine();
92.             if (s.equals("Double")){
93.                 System.out.println("Double Test");
94.                 int count=sc.nextInt();
95.                 int pop_time=sc.nextInt();
96.                 ArrayListGeneralStack arrayListGeneralStack = new ArrayListG
    eneralStack();
97.                 for (int i=0;i<count;i++){
98.                     System.out.println("push:"+arrayListGeneralStack.push(sc
    .nextDouble()));
99.                 }
100.                 for (int i=0;i<pop_time;i++){
101.                     System.out.println("pop:"+arrayListGeneralStack.pop());

102.                 }
103.                 System.out.println(arrayListGeneralStack.toString());
104.                 double sum=0;
105.                 int size=arrayListGeneralStack.size();
106.                 for (int i=0;i<size;i++){
107.                     sum+=(double)arrayListGeneralStack.pop();
108.                 }
109.                 System.out.println("sum="+sum);
110.                 System.out.println("interface GeneralStack");
111.             }else if (s.equals("Integer")){
112.                 System.out.println("Integer Test");
113.                 int count=sc.nextInt();
114.                 int pop_time=sc.nextInt();
115.                 ArrayListGeneralStack arrayListGeneralStack = new ArrayList
    GeneralStack();
116.                 for (int i=0;i<count;i++){
117.                     System.out.println("push:"+arrayListGeneralStack.push(s
    c.nextInt()));
118.                 }
119.                 for (int i=0;i<pop_time;i++){
120.                     System.out.println("pop:"+arrayListGeneralStack.pop());

121.                 }
122.                 System.out.println(arrayListGeneralStack.toString());
123.                 int sum=0;
124.                 int size=arrayListGeneralStack.size();
125.                 for (int i=0;i<size;i++){
```

```
                    sum+=(int)arrayListGeneralStack.pop();
                }
                System.out.println("sum="+sum);
                System.out.println("interface GeneralStack");
            }else if (s.equals("Car")){
                System.out.println("Car Test");
                int count=sc.nextInt();
                int pop_time=sc.nextInt();
                ArrayListGeneralStack arrayListGeneralStack = new ArrayList
GeneralStack();
                for (int i=0;i<count;i++){
                    int id=sc.nextInt();
                    String name=sc.next();
                    Car car = new Car(id,name);
                    System.out.println("push:"+arrayListGeneralStack.push(c
ar));
                }
                for (int i=0;i<pop_time;i++){
                    System.out.println("pop:"+arrayListGeneralStack.pop());

                }
                System.out.println(arrayListGeneralStack.toString());
                if (arrayListGeneralStack.size()>0){
                    int size=arrayListGeneralStack.size();
                    for (int i=0;i<size;i++){
                        Car car=(Car) arrayListGeneralStack.pop();
                        System.out.println(car.getName());
                    }
                }
                System.out.println("interface GeneralStack");
            }else if (s.equals("quit")){
                break;
            }
        }

    }
}

```