



```
1 package Composite_Pattern_1;
2
3 public abstract class FruitElement {
4     public abstract void add(FruitElement c);
5     public abstract void remove(FruitElement c);
6     public abstract FruitElement getChild(int index);
7     public abstract void eat();
8 }
9
```

```
1 package Composite_Pattern_1;
2
3 public class Apple extends FruitElement {
4
5     @Override
6     public void add(FruitElement c) {
7         // TODO 自动生成的方法存根
8         System.out.println("添加出现错误");
9     }
10
11     @Override
12     public void remove(FruitElement c) {
13         // TODO 自动生成的方法存根
14         System.out.println("删除出现错误");
15     }
16
17     @Override
18     public FruitElement getChild(int index) {
19         // TODO 自动生成的方法存根
20         System.out.println("获取子节点出现错误");
21         return null;
22     }
23
24     @Override
25     public void eat() {
26         // TODO 自动生成的方法存根
27         System.out.println("吃了一个苹果");
28     }
29
30 }
31
```

```
1 package Composite_Pattern_1;
2
3 public class Banana extends FruitElement {
4
5     @Override
6     public void add(FruitElement c) {
7         // TODO 自动生成的方法存根
8         System.out.println("添加出现错误");
9     }
10
11     @Override
12     public void remove(FruitElement c) {
13         // TODO 自动生成的方法存根
14         System.out.println("删除出现错误");
15     }
16
17     @Override
18     public FruitElement getChild(int index) {
19         // TODO 自动生成的方法存根
20         System.out.println("获取子节点出现错误");
21         return null;
22     }
23
24     @Override
25     public void eat() {
26         // TODO 自动生成的方法存根
27         System.out.println("吃了一个香蕉");
28     }
29 }
30
31
```

```
1 package Composite_Pattern_1;
2
3 public class Peer extends FruitElement {
4
5     @Override
6     public void add(FruitElement c) {
7         // TODO 自动生成的方法存根
8         System.out.println("添加出现错误");
9     }
10
11     @Override
12     public void remove(FruitElement c) {
13         // TODO 自动生成的方法存根
14         System.out.println("删除出现错误");
15     }
16
17     @Override
18     public FruitElement getChild(int index) {
19         // TODO 自动生成的方法存根
20         System.out.println("获取子节点出现错误");
21         return null;
22     }
23
24     @Override
25     public void eat() {
26         // TODO 自动生成的方法存根
27         System.out.println("吃了一个梨");
28     }
29 }
30
31 }
```

```

1 package Composite_Pattern_1;
2
3 import java.util.ArrayList;
4
5 public class FruitPlate extends FruitElement {
6     ArrayList<FruitElement> fruitElements = new ArrayList<FruitElement>();
7     @Override
8     public void add(FruitElement c) {
9         // TODO 自动生成的方法存根
10        this.fruitElements.add(c);
11    }
12
13    @Override
14    public void remove(FruitElement c) {
15        // TODO 自动生成的方法存根
16        this.fruitElements.remove(c);
17    }
18
19    @Override
20    public FruitElement getChild(int index) {
21        // TODO 自动生成的方法存根
22        return this.fruitElements.get(index);
23    }
24
25    @Override
26    public void eat() {
27        // TODO 自动生成的方法存根
28        System.out.println("开始吃果盘里的水果");
29        for(FruitElement f:this.fruitElements ) {
30            f.eat();
31        }
32    }
33
34 }

```

```

1 package Composite_Pattern_1;
2
3 public class Child {
4     //客户端类
5     FruitElement fruits;
6     public void setFruits(FruitElement fruits) {
7         this.fruits = fruits;
8     }
9     public Child(FruitElement f) {
10        this.fruits = f;
11    }
12    public void eatFruit() {
13        System.out.println("孩子开始吃水果啦! \n");
14        this.fruits.eat();
15    }
16 }

```

```

1 package Composite_Pattern_1;
2
3 public class test {
4
5     public static void main(String[] args) {
6         // TODO 自动生成的方法存根
7         FruitPlate fruitPlate = new FruitPlate();
8         //添加一些水果
9         int i=0;
10        for (i =0;i<3;i++) {
11            fruitPlate.add(new Apple());
12        }
13        for (i =0;i<2;i++) {
14            fruitPlate.add(new Banana());
15        }
16        for (i =0;i<4;i++) {
17            fruitPlate.add(new Peer());
18        }
19
20        //添加一些水果
21        FruitPlate fruitPlate2 = new FruitPlate();
22
23
24        for (i =0;i<1;i++) {
25            fruitPlate2.add(new Apple());
26        }
27        for (i =0;i<3;i++) {
28            fruitPlate2.add(new Banana());
29        }
30        for (i =0;i<2;i++) {
31            fruitPlate2.add(new Peer());
32        }
33        fruitPlate.add(fruitPlate2);
34
35        Child child = new Child(fruitPlate);
36        child.eatFruit();
37    }
38
39 }

```

控制台

<已终止> test (4) [Java 应用程序] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (2019年10月14日 下午1:51:30)

孩子开始吃水果啦!

开始吃果盘里的水果

吃了一个苹果

吃了一个苹果

吃了一个苹果

吃了一个香蕉

吃了一个香蕉

吃了一个梨

吃了一个梨

吃了一个梨

吃了一个梨

开始吃果盘里的水果

吃了一个苹果

吃了一个香蕉

吃了一个香蕉

吃了一个香蕉

吃了一个梨

吃了一个梨