NJUSWI
2002-2012
Anniversary
南京大學 软件学院

避免重复

刘 钦

南京大学软件学院

知道程序员最喜欢的2个操作是什么吗？

ctrl c

ctrl v

重复输出

ctrl c

ctrl c

ctrl c

代码重复

```
System.out.println(“ctrl c ”);
System.out.println(“ctrl c ”);
System.out.println(“ctrl c ”);
System.out.println(“ctrl c ”);
```

重复输出

ctrl c ctrl c ctrl c ctrl c

重复

```
System.out.println(“ctrl c”);
System.out.println(“ctrl c”);
System.out.println(“ctrl c”);
System.out.println(“ctrl c”);
System.out.println(“ctrl c”);
System.out.println(“ctrl c”);
```

消除重复

- for(i=0;i<4;i++)
- System.out.print(“ctrl c”);

避免重复

重复代码

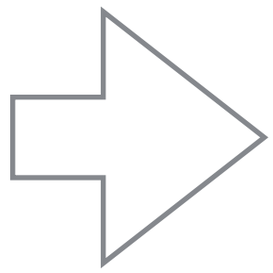
```
for (int i = 0; i < 4;
i++)
{
    sum1 += array1[i];
}
average1 = sum1/4;

for (int i = 0; i < 4;
i++)
{
    sum2 += array2[i];
}
average2 = sum2/4;
```

消除重复 — 抽象

```
for (int i = 0; i < 4;
i++)
{
   sum1 += array1[i];
}
average1 = sum1/4;

for (int i = 0; i < 4;
i++)
{
   sum2 += array2[i];
}
average2 = sum2/4;
```



```
int calcAverage (int* Array_of_4)
{
   int sum = 0;
   for (int i = 0; i < 4; i++)
   {
      sum += Array_of_4[i];
   }
   return sum/4;
}

int average1 = calcAverage(array1);
int average2 = calcAverage(array2);
```

```
int calcAverage (int* Array_of_4)
{
   int sum = 0;
   for (int i = 0; i < 4; i++)
   {
       sum += Array_of_4[i];
   }
   return sum/4;
}
```

```
int average1 = calcAverage(array1);
int average2 = calcAverage(array2);
```

消除重复 — 抽象

```
private Double getTotalSum(List amounts) {
    double totalToPay = 0.00;
    Iterator amountsIterator = amounts.iterator();
    while (amountsIterator.hasNext()) {
        Amount amount = (Amount) amountsIterator.next();
        if (!cancelstatuses.contains(amount.getStatus())) {
            totalToPay += amount.doubleValue();
        }
                                    }
    return new Double(totalToPay);
}
```

```
private Double getTotalSumExcludeCancelAmount(List amounts) {
    double totalToPay = 0.00;
    Iterator amountsIterator = amounts.iterator();
    while (amountsIterator.hasNext()) {
        Amount amount = (Amount) amountsIterator.next();
        if (!amount.getIsToCancel()) { // Additional condition comparing to the first method.
            if (!cancelstatuses.contains(amount.getStatus())) {
                totalToPay += amount.doubleValue();
            }
        }
                                                }
    return new Double(totalToPay);
}
```