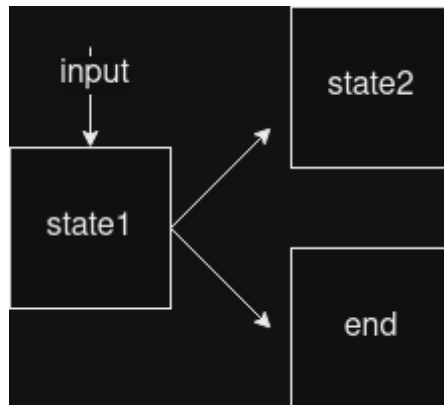


自动状态机

1. 设置系统状态
2. 设置输入
3. 设置状态转移函数/map



1 设置系统状态

```
1 enum State{
2     STATE_1,
3     STATE_2,
4     STATE_3
5 }
```

2 设置输入类型

可以直接使用普通参数设置输入，也可以将参数转换成设定的输入类型

```
1 enum InputType{
2     InputType_1,
3     InputType_2,
4     InputType_3
5 }
```

```

6      InputType toInputType(auto args,...){
7          // if或case 判断语句，将输入参数转换格式
8          if(...){
9              return InputType_1;
10         }else if(...){
11             return InputType_2;
12         }else{
13             return InputType_3;
14         }
15     }

```

3 设置状态转移方法

```

1  //使用函数
2      State switchState(State state, InputType
inputtype){
3          // if或case 判断语句，将输入参数转换格式
4          State stateRes;
5          if(state == state1){
6              if(inputtype == InputType_1){
7                  stateRes = state2;
8              }
9              else if(inputtype == InputType_2){
10                 ...
11             }
12         }
13         else if(state == state2){
14             ...
15         }else stateRes = state3;
16         return stateRes;
17     }
18
19 // 使用map
20 // 其实就是字典

```

```

21     unordered_map<State,
    unordered_map<InputType, State>> tranfer{
22         {
23             State1,
24             {
25                 {InputType_1, State1},
26                 {InputType_2, State2}
27             }
28         },
29         ...
30     }

```

状态转移方法无论是使用什么途径，遵循的公式都是

$$\text{状态1} + \text{输入参数} = \text{状态2} \quad (1)$$

使用函数进行状态转移，是将 **当前状态** 和 **输入参数** 作为参数，返回 **转移后的状态**

使用 map 数据结构进行状态转移，主要是通过将 **当前状态** 和 **输入参数** 作为两个关键字，然后查询的值为 **转移后的状态**

4 使用

1. 初始化状态机为初始状态
2. 挨个将输入数据读入状态机中
3. 对最后得到的状态机状态进行判断，得出最终结果

5 设置状态的技巧

1. 首先根据系统状态改变的流程设置，根据系统流程，将系统拆解成若干单元，将每一单元设置成状态
2. 根据输入输出设置，在第一步的基础上，我们列举每一个单元的输入输出，将相同输入输出的单元合并

