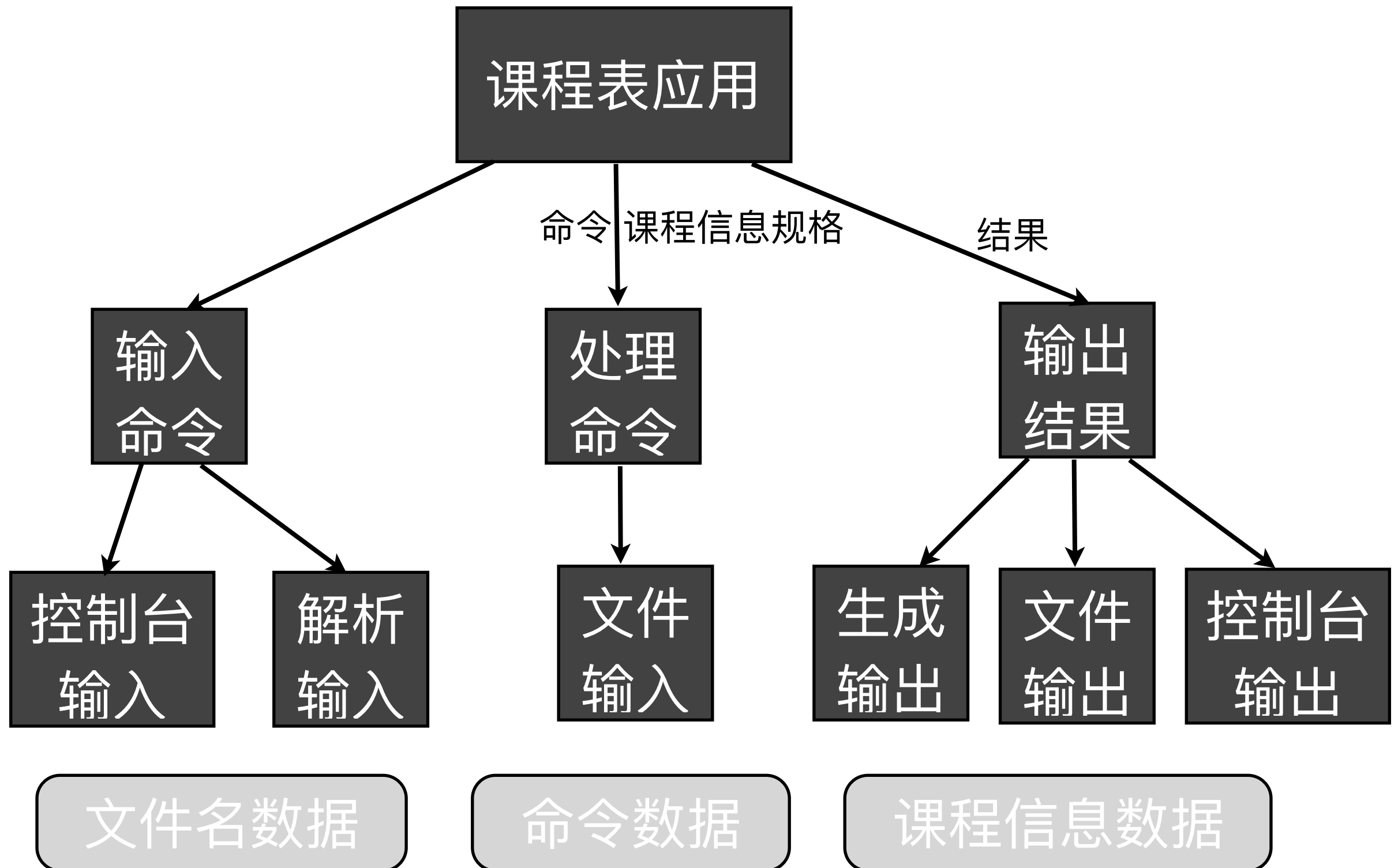

大范围的修改
VS
有限范围的修改

让我们来看看结构化
方法的问题

课程表案例

- 建一个课程表。
 - 星期四；三，四节；计算与软件工程；仙2-407；
- 通过命令行方式完成对课程的增、删、改、查、显示。
 - Add 星期四；三，四节；计算与软件工程；仙2-407； //如果成功 显示“已添加到文件中”
 - Remove 星期四；三，四节；计算与软件工程；仙2-407； //如果成功 显示“已从文件删除”
 - Update 星期四；三，四节；计算与软件工程；仙2-408； //如果成功 显示“已更新文件”
 - Find 星期四；三，四节； //如存在课程 显示“课程名；上课地点”
 - Show //显示所有课程，按照时间排序
- 数据保存在文件里。
 - CurriculumSchedule.txt

课程表应用回顾--结构化设计



Problems of Structured Programming

- Not easy to read
- Not easy to maintain

Not Easy to Read -- 全局变量

```
public class CourseSchedule {
```

```
    static public String fileName = "CurriculumSchedule";
```

第11行

```
    public static void main(String[] args){  
        String input = "", output = "";  
        String command;  
        String courseInfo;  
        int cmd = -1;
```

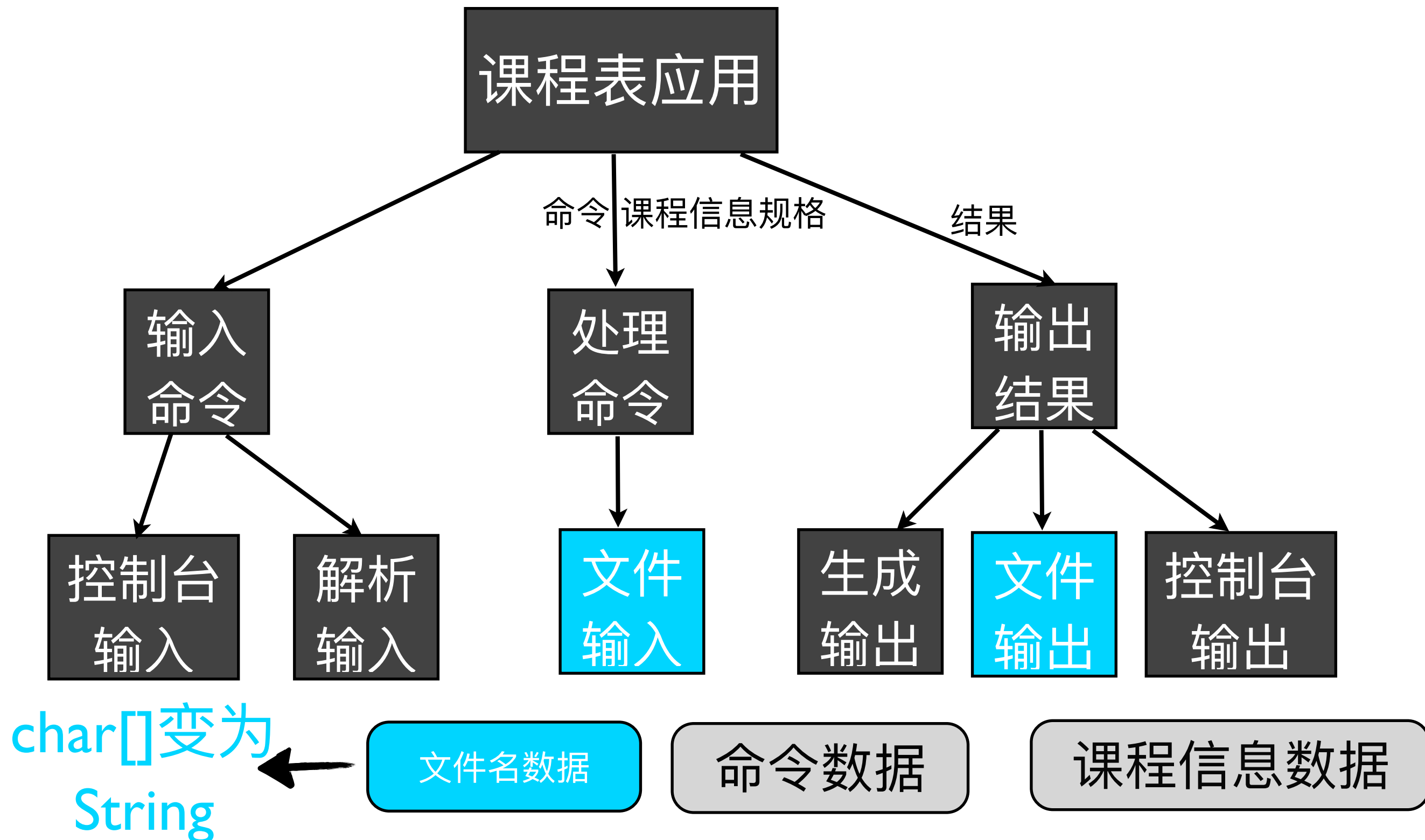
```
//        System.out.println("courseInfo:"+day+time+name+location);  
        try{  
            BufferedReader br1=new BufferedReader(new FileReader(fileName));  
            String line;  
            while((line=br1.readLine())!=null){  
                String day2;  
                String time2;  
                String name2;  
                String location2;
```

第137行

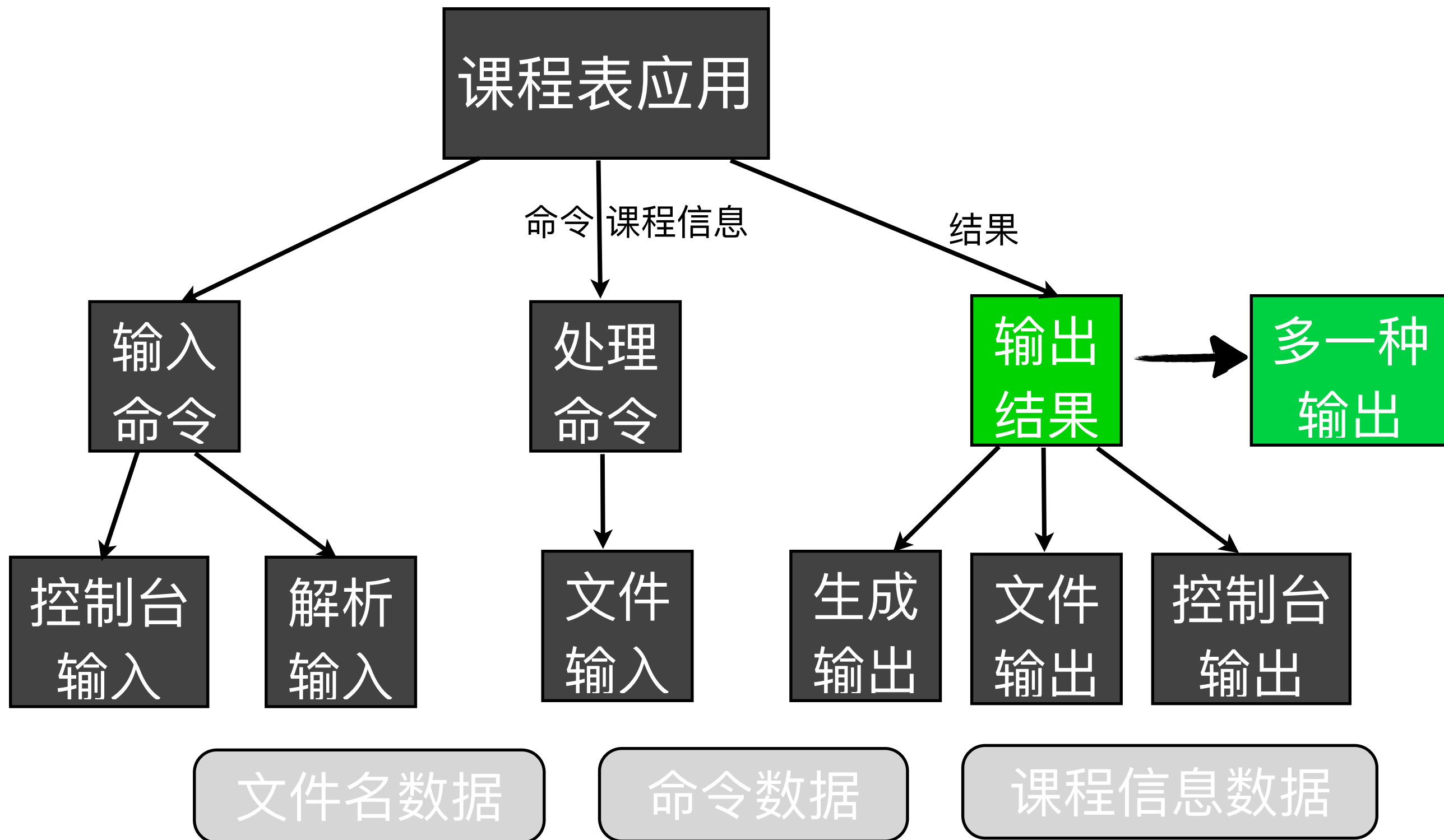
第237行

...

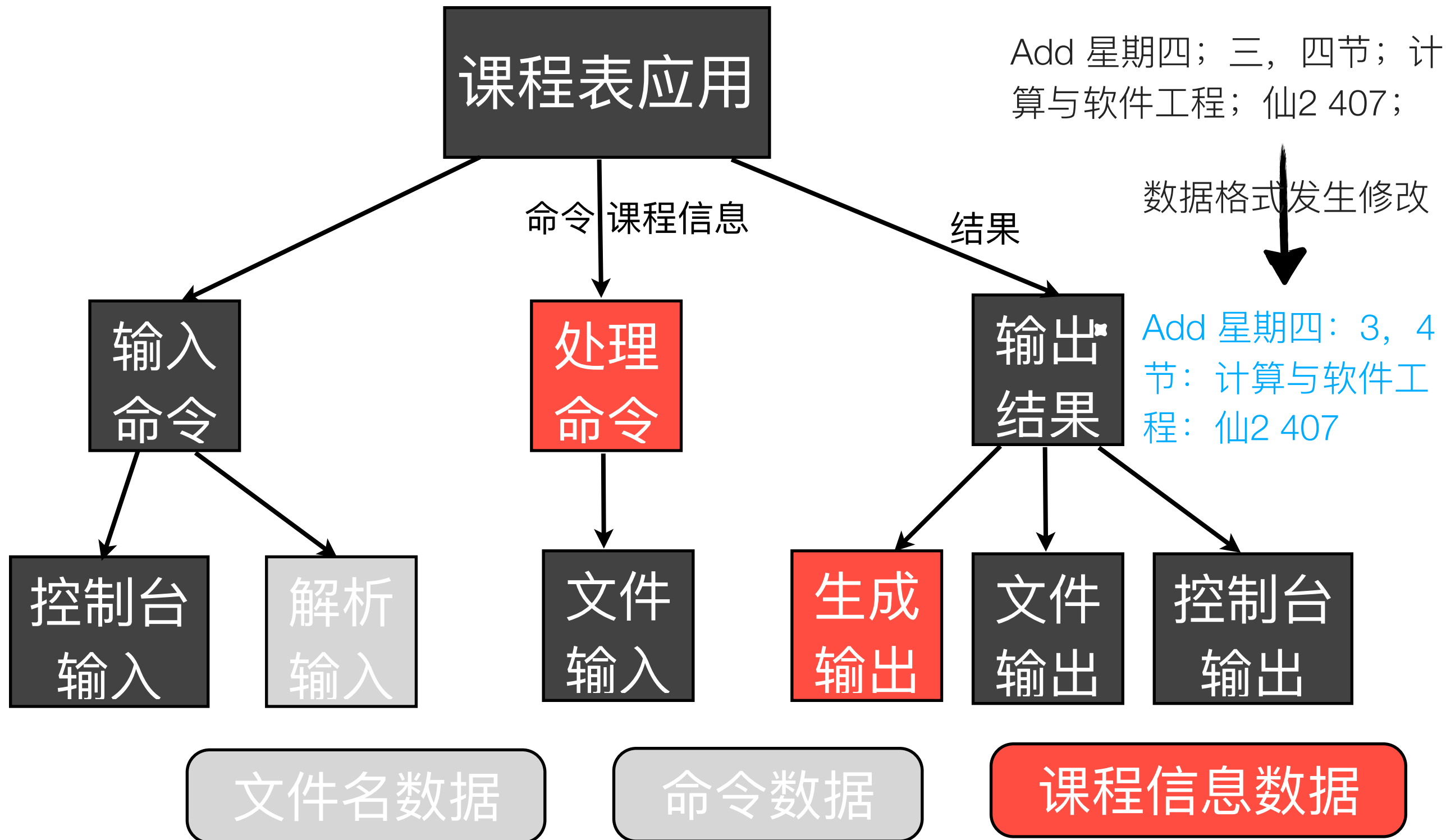
Not Easy to Maintain -- 实现变更



Not Easy to Maintain -- 需求增加



Not Easy to Maintain -- 需求更改



大范围的修改

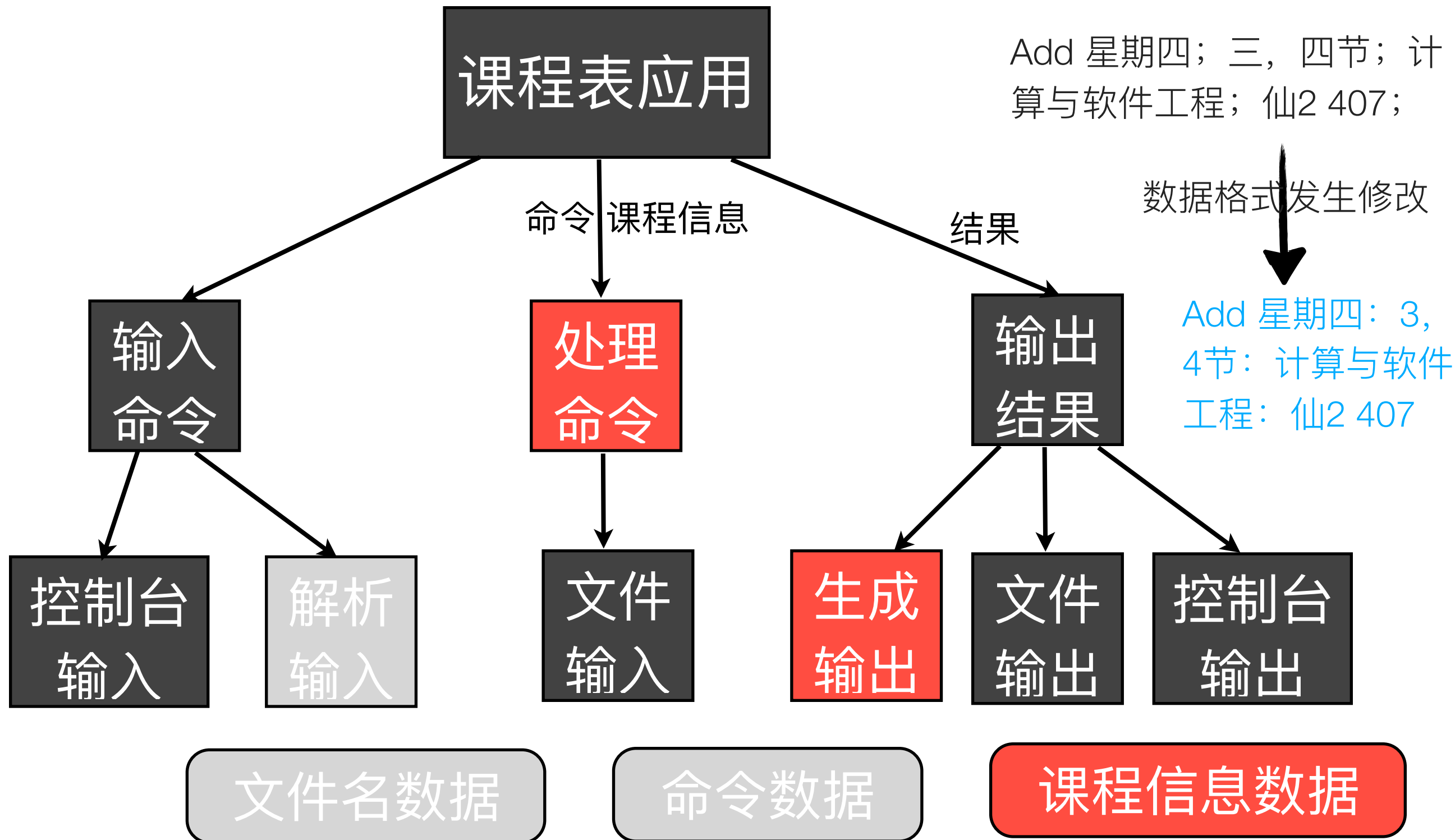
Nightmare!

问题

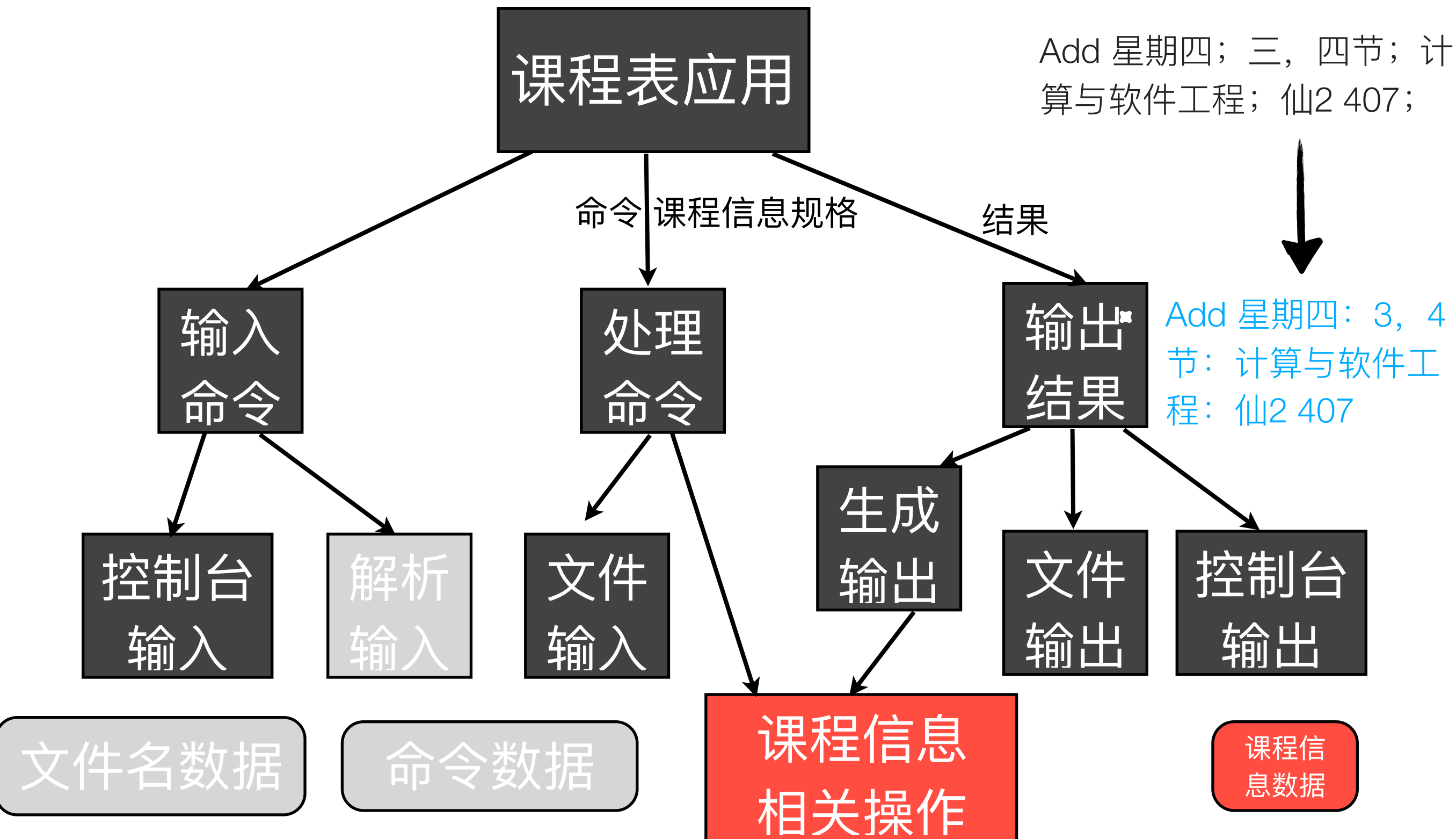
Q: 如果我们不希望大范围的修改,
那么怎么应对变更?

A: 1. 在有限的范围内修改
2. 扩展

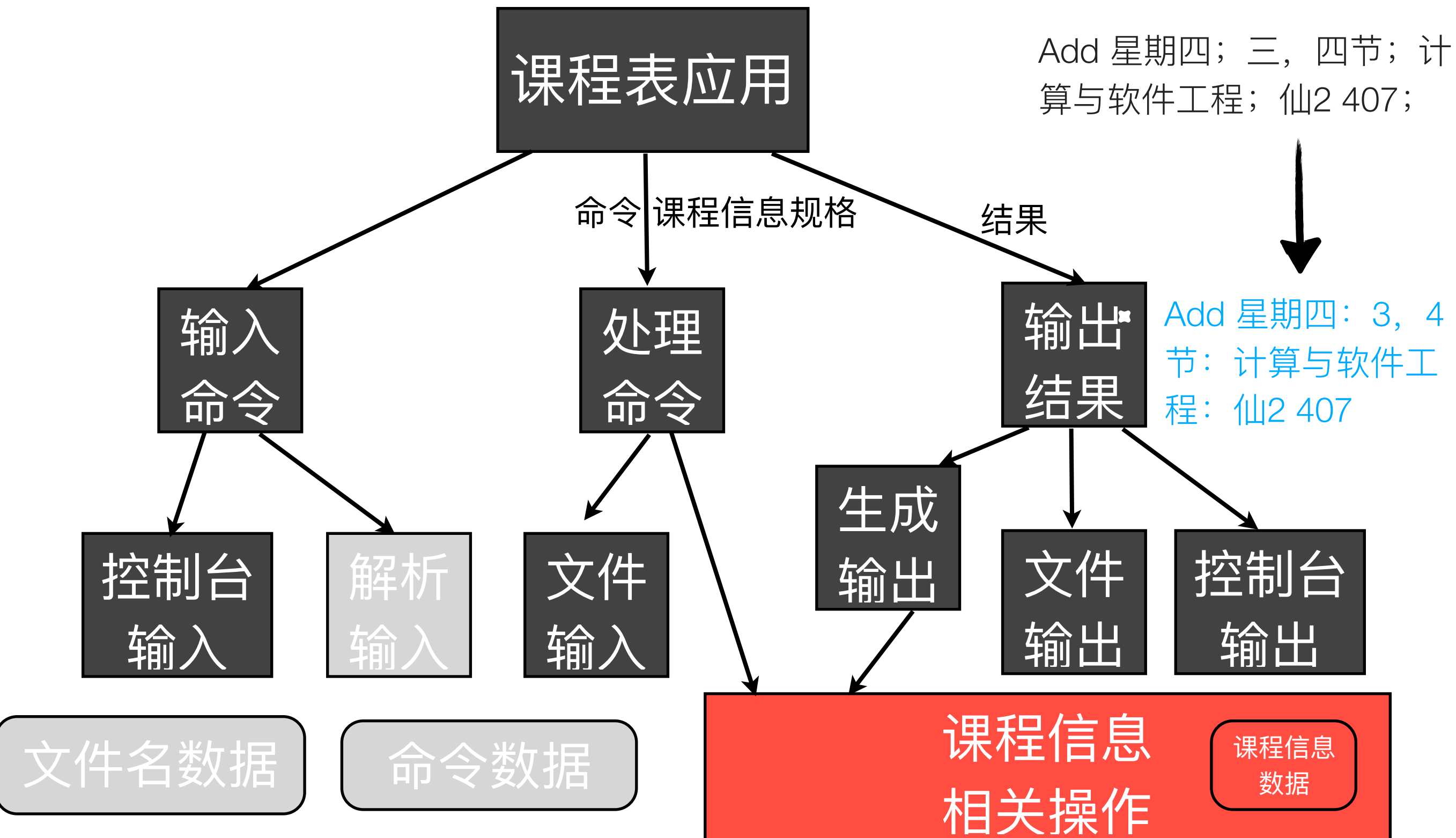
大范围的修改



在有限的范围内修改--相关操作在一起



在有限的范围内修改--数据与操作在一起（封装）



扩展—运行时动态链接（继承、多态）

