



自顶向下的设计

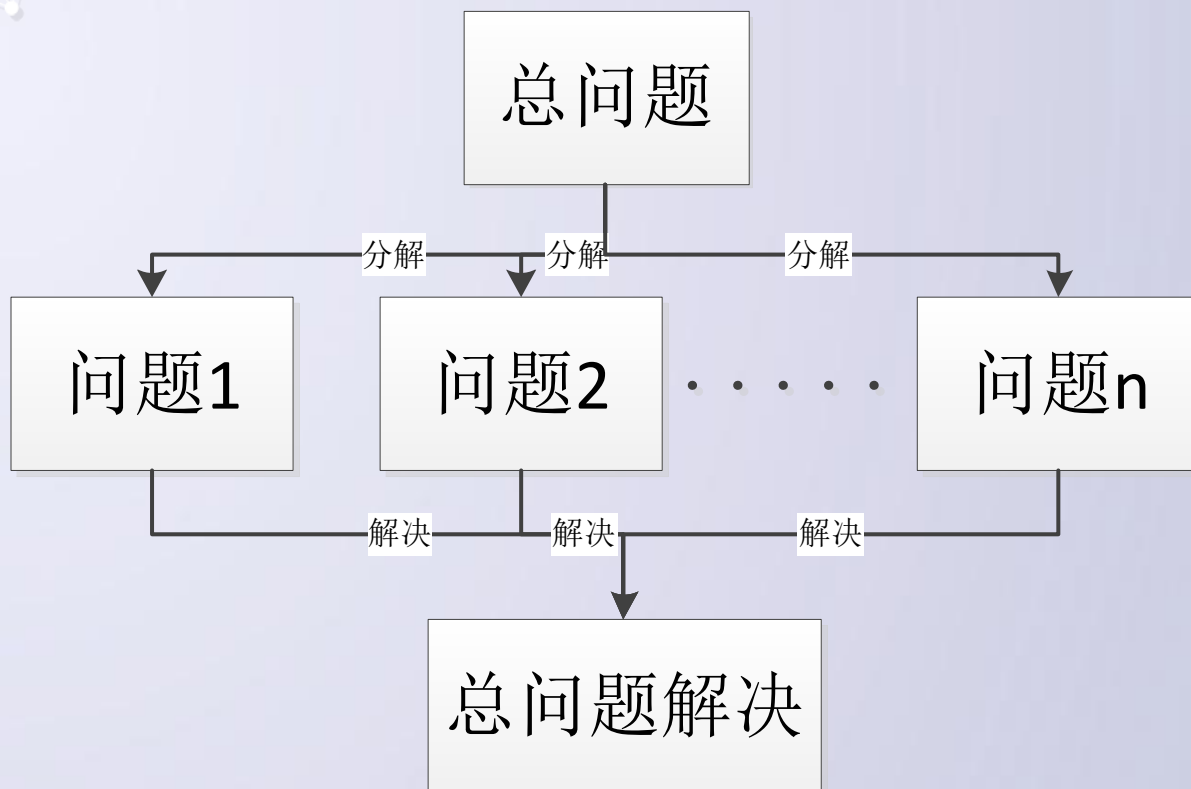
黄天羽

北京理工大学





基本思想





顶层设计（第一阶段）举例

- 步骤1：打印程序的介绍性信息

```
printIntro()
```

- 步骤2：获得程序运行所需的参数：ProA、ProB、n

```
probA,probB,n = getInputs()
```



顶层设计（第一阶段）举例

- 步骤3：模拟n次比赛

```
winsA, winsB = simNGames(n,probA,probB)
```

- 步骤4：输出球员A和B获胜比赛的次数和概率

```
printSummary(winsA, winsB)
```



顶层设计（第一阶段）举例

```
def main():
```

```
    printIntro()
```

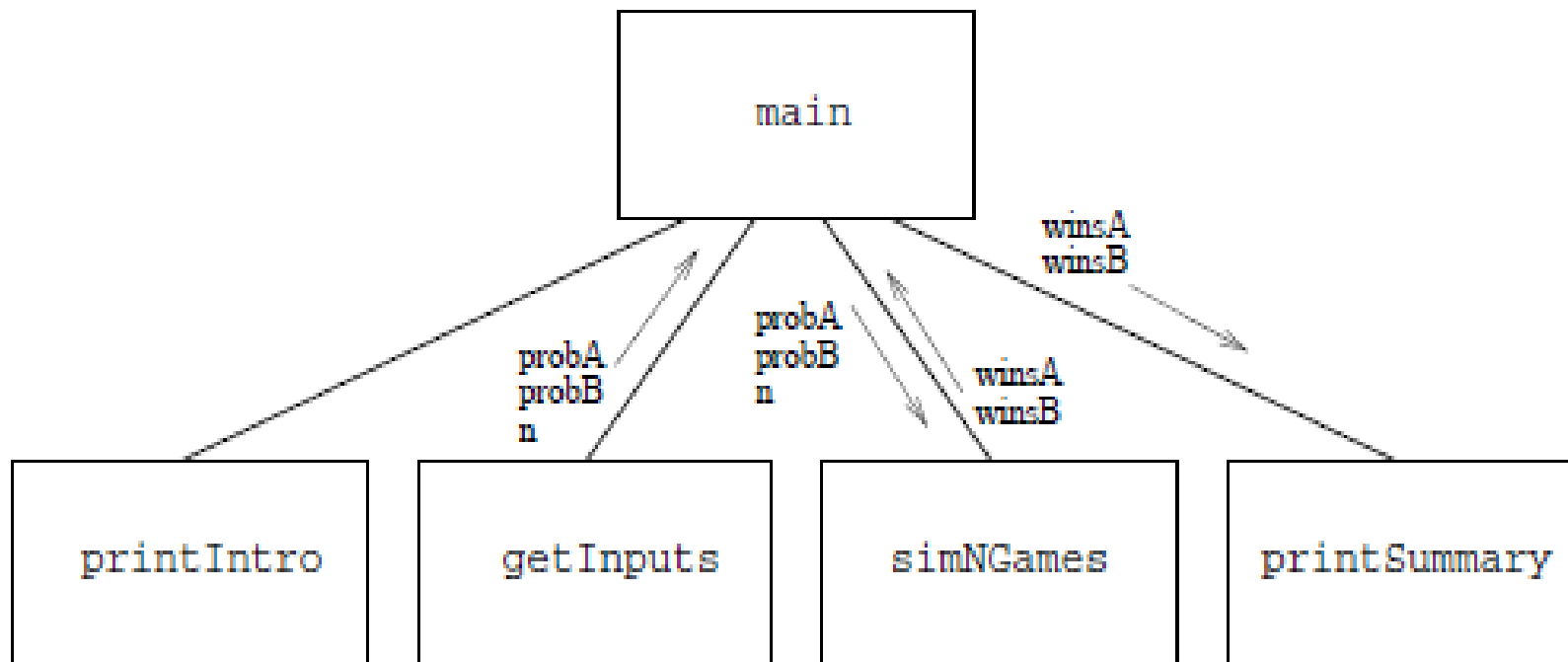
```
    probA,probB,n = getInputs()
```

```
    winsA, winsB = simNGames(n,probA,probB)
```

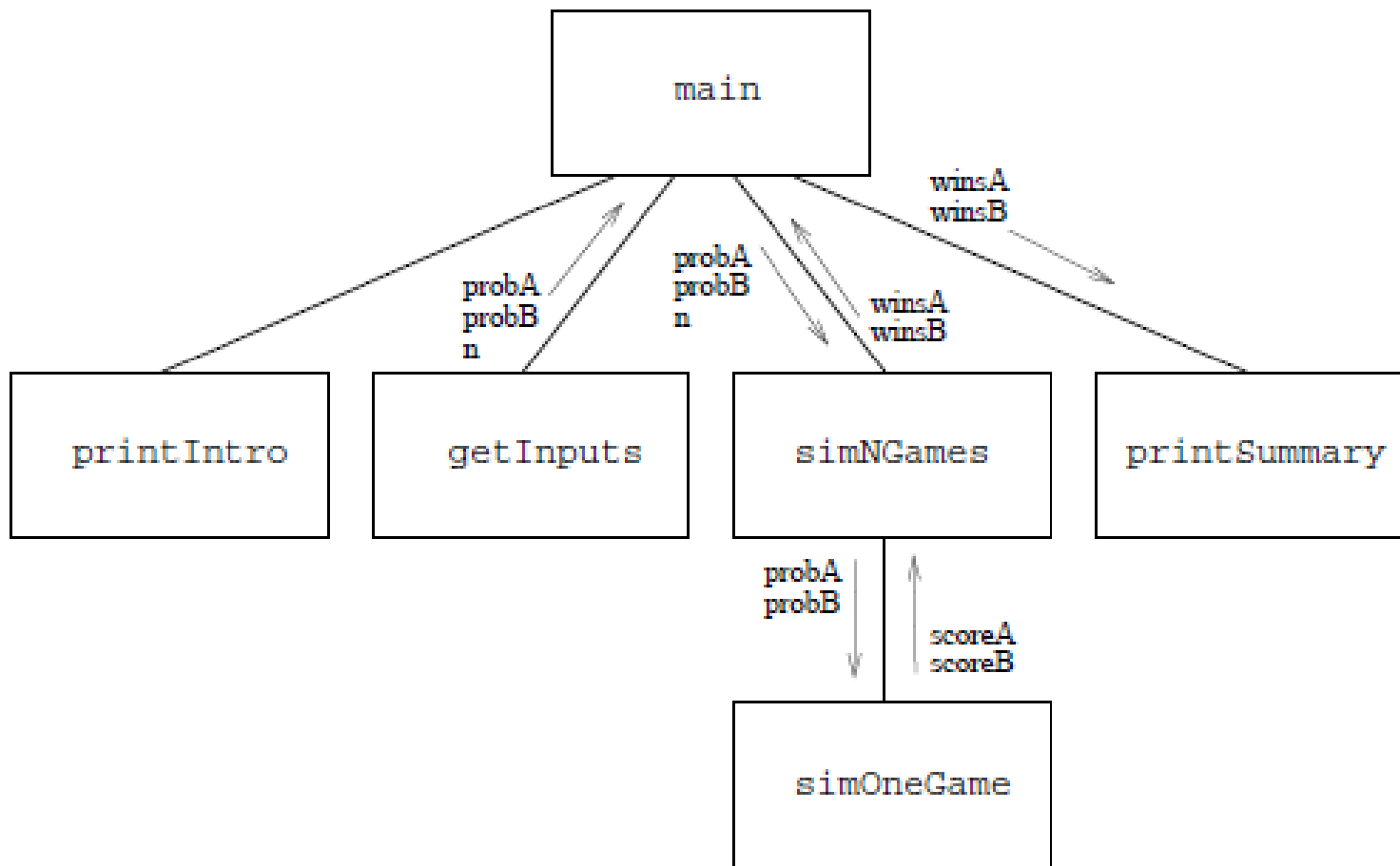
```
    PrintSummary(winsA, winsB)
```



第一阶段



第二阶段



语言程序设计 体育竞技分析程序结构图：第二阶段



第二阶段

■ PrintIntro()函数；

```
def printIntro():
```

```
    print( 'This program simulates a game between two' )
```

```
    print( 'There are two players, A and B' )
```

```
    print( 'Probability(a number between 0 and 1)is used' )
```





第二阶段

- `getInputs()`函数；

```
def getInputs():
```

```
    a = eval(input( 'What is the prob.player A wins?' ))
```

```
    b = eval(input( 'What is the prob.player B wins?' ))
```

```
    n = eval(input( 'How many games to simulate?' ))
```

```
    return a,b,n
```



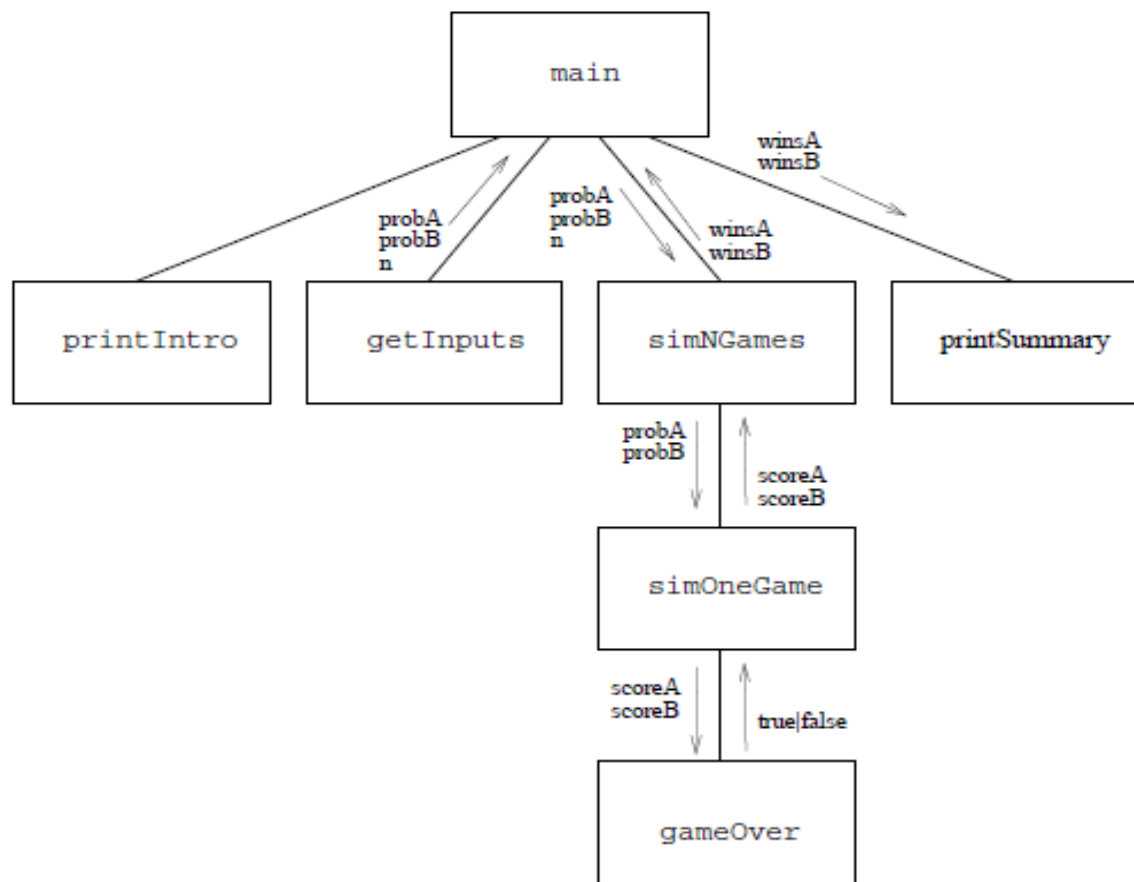
第二阶段

■ simNGames()函数(核心)

```
def simNGames(n,probA,probB):  
    winsA = 0  
    winsB = 0  
    for i in range(n):  
        scoreA,scoreB = simOneGame(probA,probB)  
        if scoreA > scoreB:  
            winsA = winsA + 1  
        else:  
            winsB = winsB + 1  
    return winsA,winsB
```



第三阶段





第三阶段

- `simOneGame()`函数；

```
def simOneGame(probA,probB):
```

```
    scoreA = 0
```

```
    scoreB = 0
```

```
    serving = 'A'
```

```
    while <condition>:
```

```
        <todo>
```





第三阶段

■ simOneGame()函数完整代码：

```
def simOneGame(probA,probB):  
    scoreA = 0  
    scoreB = 0  
    serving = "A"  
    while not gameOver(scoreA,scoreB):  
        if serving == "A":  
            if random() < probA:  
                scoreA = scoreA + 1  
            else:  
                serving = "B"  
        else:  
            if random() < probB:  
                scoreB = scoreB + 1  
            else:  
                serving = "A"  
    return scoreA,scoreB
```





第三阶段

- gameOver()函数：

```
def gameOver(a,b):  
    return a==15 or b==15
```

- printSummary()函数

```
def printSummary():  
    n = winsA + winsB  
    print( '\nGames simulated:%d' %n)  
    print( 'WinsforA:{0}({1:0.1%})'.format(winsA,winsA/n))  
    print( 'Wins for B:{0}({1:0.1%})'.format(winsB,winsB/n))
```





设计过程总结

- 自顶向下设计：
 - 步骤1：将算法表达为一系列小问题；
 - 步骤2：为每个小问题设计接口；
 - 步骤3：通过将算法表达为接口关联的多个小问题来细化算法；
 - 步骤4：为每个小问题重复上述过程。