**数据结构实验报告14**

**学号：** 117060400211 **姓名**：王婕冉 **班级：** 应用统计学2班 **指导老师：林卫中**

**实验名称**： 程序设计方法论

**实验要求：** 了解计算思维的概念

掌握自顶向下的设计方法

掌握自底向上的执行过程

**实验题目：模拟多个比赛，分析其规律**

**算法实现：**

**Ⅰ：乒乓球规则模拟**

from random import random

def printIntro():

    print("这个程序模拟两个选手A和B的某种竞技比赛")

    print("程序运行需要A和B的能力值（以0到1之间的小数表示）")

def getInputs():

    a = eval(input("请输入选手A的能力值(0-1): "))

    b = eval(input("请输入选手B的能力值(0-1): "))

    n = eval(input("模拟比赛的场次: "))

    return a, b, n

def simNGames(n, probA, probB):

    winsA, winsB = 0, 0

    for i in range(n):

        scoreA, scoreB = simOneGame(probA, probB)

        if scoreA > scoreB:

            winsA += 1

        else:

            winsB += 1

    return winsA, winsB

def gameOver(a,b):

    return a==11 or b==11

def simOneGame(probA, probB):

    scoreA, scoreB = 0, 0

    serving = 0

    t = 0

    while not gameOver(scoreA, scoreB):

        if serving == 0:

            if random() < probA:

                scoreA += 1

            else:

                scoreB += 1

        else:

            if random() < probB:

                scoreB += 1

            else:

                scoreA += 1

        t += 1

        if t%2 == 0:

            serving = (serving+1)%2

    return scoreA, scoreB

def printSummary(winsA, winsB):

    n = winsA + winsB

    print("竞技分析开始，共模拟{}场比赛".format(n))

    print("选手A获胜{}场比赛，占比{:0.1%}".format(winsA, winsA/n))

    print("选手B获胜{}场比赛，占比{:0.1%}".format(winsB, winsB/n))

def main():

    printIntro()

    probA, probB, n = getInputs()

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

    printSummary(winsA, winsB)

main()

或改进规则：

def gameOver(a,b):

    if (a==11 and b<=9) or (b==11 and a<=9):

        return True

    elif (30>a>=10 and b-a==2) or (30>b>=10 and a-b==2):

        return True

    elif a==30 or b==30:

        return True

**Ⅱ：篮球规则模拟比赛**

from random import random

from random import randint

def printIntro():

    print("这个程序模拟两个选手A和B的某种竞技比赛")

    print("程序运行需要A和B的能力值（以0到1之间的小数表示）")

def getInputs():

    g1 = eval(input("请输入球队A的投篮能力值(0-1): "))

    b1 = eval(input("请输入球队A的篮板能力值(0-1): "))

    g2 = eval(input("请输入球队B的投篮能力值(0-1): "))

    b2 = eval(input("请输入球队B的篮板能力值(0-1): "))

    n = eval(input("请输入比赛的场数: "))

    return g1, b1, g2, b2, n

def simNGames(n, goleA, boardA, goleB, boardB):

    winsA, winsB = 0, 0

    for i in range(n):

        scoreA, scoreB = simOneGame(goleA, boardA, goleB, boardB)

        if scoreA > scoreB:

            winsA += 1

        else:

            winsB += 1

    return winsA, winsB

def gameOver(t):

    return t >=12\*60

def simOneGame(goleA, boardA, goleB, boardB):

    scoreA, scoreB = 0, 0

    serving = 0

    totalTime = 0

    while not gameOver(totalTime):

        t = randint(1, 24)

        totalTime += t

        if t == 24:

            serving == (serving +1)%2

        else:

            if serving == 0:

                if random() < goleA:

                    scoreA += 1

                    serving = 1

                else:

                    if random() < boardA:

                        serving = 0

                    else:

                        serving = 1

            else:

                if random() < goleB:

                    scoreB += 1

                    serving = 0

                else:

                    if random() < boardB:

                        serving = 1

                    else:

                        serving = 0

    return scoreA, scoreB

def printSummary(winsA, winsB):

    n = winsA + winsB

    print("竞技分析开始，共模拟{}场比赛".format(n))

    print("选手A获胜{}场比赛，占比{:0.1%}".format(winsA, winsA/n))

    print("选手B获胜{}场比赛，占比{:0.1%}".format(winsB, winsB/n))

def main():

    printIntro()

    goleA, boardA, goleB, boardB, n = getInputs()

    winsA, winsB = simNGames(n, goleA, boardA, goleB, boardB)

    printSummary(winsA, winsB)

main()

**实验结果：**



