

201901 学期《Java 程序设计》线下考试答案及评分标准（A 卷）

一. 卷选择题答案:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	D	A	C	A	D	D	B	A	D	B	D	D	A	C	D	D	B	C	A

二. 判断题答案:

1	2	3	4	5	6	7	8	9	10
×	√	×	√	√	√	×	×	√	√

三. 完善程序填空题（每空 2 分，每题 5 空）

给定学生类型，要求完善程序以实现按学生的 age 字段由大到小排序：

```
class Student implements Comparable<Student> {
    String name; int age;
    Student(String n, int a) { name = n; age = a; }
    public int compareTo(Student s) {
        __return s.age - age__;
    }
}

public class TreeSetSort {
    public static void main(String[] args) {
        TreeSet<Student> t = new TreeSet<Person>();
        __t.add(new Student("张三", 20))__; //增加一个学生：姓名张三，年龄 20
        __t.add(new Student("李四", 21))__; //增加一个学生：姓名李四，年龄 21
        Iterator< Student> it = t.iterator();
        While( __it.hasNext() __) {
            System.out.println( it.next() );
        }
    }
}
```

实现一个线程，输出 k=1..100 之内的所有整数，每输出一个 k，睡眠[1,1000]以内的随机毫秒。

```
import java.util.Random;
class GenDigit extends __Thread__ {
    public void run() {
        for (int k=1; k<= 100; ++k) {
```

```

        System.out.println( __k__ );
    try {
        Random r = new Random();
        Thread. sleep( 1 + r.nextInt(1000) );
    }
    catch( Exception e ) {}
}
}
}

```

四 、 程序设计题（每小题 10 分，共 30 分）

程序设计题 1:

主要代码和参考评分标准：每行 2 分，共 10 分；

```

int n=data.length;
int pos=n/2;
double result;
if (n%2==0) result=(double)(data[pos]+data[pos-1])/2;
else result=data[pos];

```

程序设计题 2:

主要代码和参考评分标准:

有 CPU（必）、Memory（必）、Bus（可选）和 Computer（可选）等类；（4 分）

CPU 类有读、写、计算（如加）等方法（读、写是通过 Bus 访问 Memory）；Memory 有读、写的方法及实现。（6 分）

```

public class CPU {
    private Bus bus;
    public CPU(){}
    public void connectBus(Bus aBus) {
        bus=aBus;
    }
    public void add(int addr1, int addr2, int addr3) {
        int a =bus.read(addr1);
        int b= bus.read(addr2);
        bus.write(a+b, addr3);
        return;
    }
    public void assign(int value, int addr) {
        bus.write(value, addr);
    }
}

```

```

        public void show(int addr) {
            System.out.println(bus.read(addr));
        }
    }

    public class Memory {
        private int data[];
        Memory() {
            data=new int[100];
        }
        public boolean write(int value, int addr) {
            data[addr]=value;
            return true;
        }
        public int read(int addr) {
            return data[addr];
        }
    }

    public class Bus {
        private Memory mem;
        private CPU cpu;
        public Bus() {};
        public void connectCPU(CPU acpu) {
            cpu=acpu;
        }
        public void connectMemory(Memory mem) {
            this.mem=mem;
        }
        public boolean write(int value, int addr) {
            return mem.write(value, addr);
        }
        public int read(int addr) {
            return mem.read(addr);
        }
    }

    public class Computer {
        public static void main(String args[]) {
            CPU aCPU=new CPU();
            Bus aBus=new Bus();
            Memory aMem=new Memory();

            aCPU.connectBus(aBus);
            aBus.connectCPU(aCPU);
            aBus.connectMemory(aMem);
        }
    }

```

```

        aCPU.assign(5, 1);
        aCPU.assign(4, 2);
        aCPU.add(1,2,3);
        aCPU.show(3);
    }
}

```

程序设计题 3:

主要代码和参考评分标准:

设计了角色控制器 (**Controller**)、抽象角色类 (或接口 **IRole**) 和具体角色类 (**UserDefinedRole**) 等类 (3 分)

Controller 中, 声明 **IRole aRole=new UserDefinedRole()**, 后续使用 **IRole** 编写代码。

IRole 中声明了展示、走、跑、跳、爬等方法但不实现或有缺省实现; (4 分)

UserDefinedRole 继承 **IRole** 并实现几个方法。(3 分)

其他方式、方法可实现类似效果的, 可酌情给分。

利用设计模式如 **Builder** 模式等, 将 **UserDefinedRole** 拆为由不同部分构成的, 可酌情加分。

五 、 附加题 (30 分)

```

class Array {
public int[] p;
public int m, int n;
public void init(int n) {
    this.n = n; p = new int[n]
}

    void init(int m, int n) {
        this.m = m; this.n = n;
        p = new int[m * n];
    }
    int at(int i) { return p[i]; }
    void set(int i, int d) { p[i]= d; }
    int at(int i, int j) { return p[i*n+j]; }
    void set(int i, int j, int d) { p[i*n+j]=d; }
}; //5 分

class OpBST {

```

```

public:
    int n;          //how many keys
    Array A= new Array();//      A_1, A_2, ..., A_n:
frequencies
    Array B= new Array();//B0 B_1, B_2, ..., B_n
    Array key= new Array(); //      k_1, k_2, .....k_n
    Array cost= new Array(); //n+2*n+1: cost(i,j): i in
[1,n+1], j in [0,n]
    Array w=new Array(); //n+2*n+1: w_i,j = w_i,j-1
+ A_j + B_j, i in [1,n+1], j in [0,n]
    Array root=new Array(); //n+1*n+1: root(i,j) of the
problem key[i..j], i in [1..n], j[1..n], i<=j
    bool isFirst;
    Scanner scn = new Scanner();
    OpBST() {isFirst = true; }
    void readN() {
        n = s.nextInt();
        A.init(n+1);
        B.init(n+1);
        key.init(n+1);
        cost.init(n+2, n+1);
        w.init(n+2, n+1);
        root.init(n+1, n+1);
    }
    void readKeyAB() {
        int i=1;
        for(; i<=n; ++i) {
            key.set(i, s.nextInt());
        }
        for(i=1; i<=n; ++i) {
            A.set(i, s.nextInt());
        }
        B.set(0, s.nextInt());
        for(i=1; i<=n; ++i) {

```

```

        B.set(i, s.nextInt());
    }
    //init cost matrix
    //cost[i,i-1] = W[i,i-1] = B[i-1]   for 1<= i <= n+1
    for(i=1; i<=(n+1); ++i) {
        int d = B.at(i-1);
        cost.set(i, i-1, d);
        w.set(i, i-1, d);
    } //10 分
}

void findMinIn(int i, int j) {
    int min = 0x7FFFFFFF;
    int rt = -1;
    int r=i;
    for(; r<=j; ++r) {
        int v = cost.at(i, r-1)+cost.at(r+1, j) +
w.at(i,j);
        if (v<min) {
            min = v;
            rt = r;
        }
    }
    Cost.set(i,j, min);
    root.set(i,j, rt); //5 分
}

void computeSol() {
    //W[i,j]=W[i,j-1]+A[j]+B[j]
    //cost[i,i-1] = min_i<=r<=j:
{cost[i,r-1]+cost[r+1,j]+W[i,j])   for 1<=i <= j<=n
    int k=0;
    for(; k<n; ++k) {
        int i;
        for (i = 1; i <= n; ++i) {
            int j=i+k;

```

```

        // w_i,j = w_i,j-1 + A_j + B_j, i in [1,n+1],
j in [0,n]
        int n = w.at(i, j - 1) + A.at(j) + B.at(j);
        w.set(i, j, n)
        findMinIn(i,j); //10 分
    }
}
}

void printOBST(int i, int j) {
    if (j < i) return;
    //in preorder
    int r = root.at(i,j);
    if (!isFirst) {
        System.out.print(" ");
    }
    System.out.print(key.at(r)); //root
    if (isFirst) {
        isFirst = false;
    }
    printOBST(i, r-1); //left subtree
    printOBST(r+1, j); //right subtree
}

void main() {
    readN();
    readKeyAB();
    computeSol();
    System.out.println(cost.at(1,n));
    printOBST(1,n);
}

};

public static void main(String[] args){
    OpBST b = new OpBST();
    b.main();
}

```