# 2018年下半年程序员考试下午真题（参考答案）

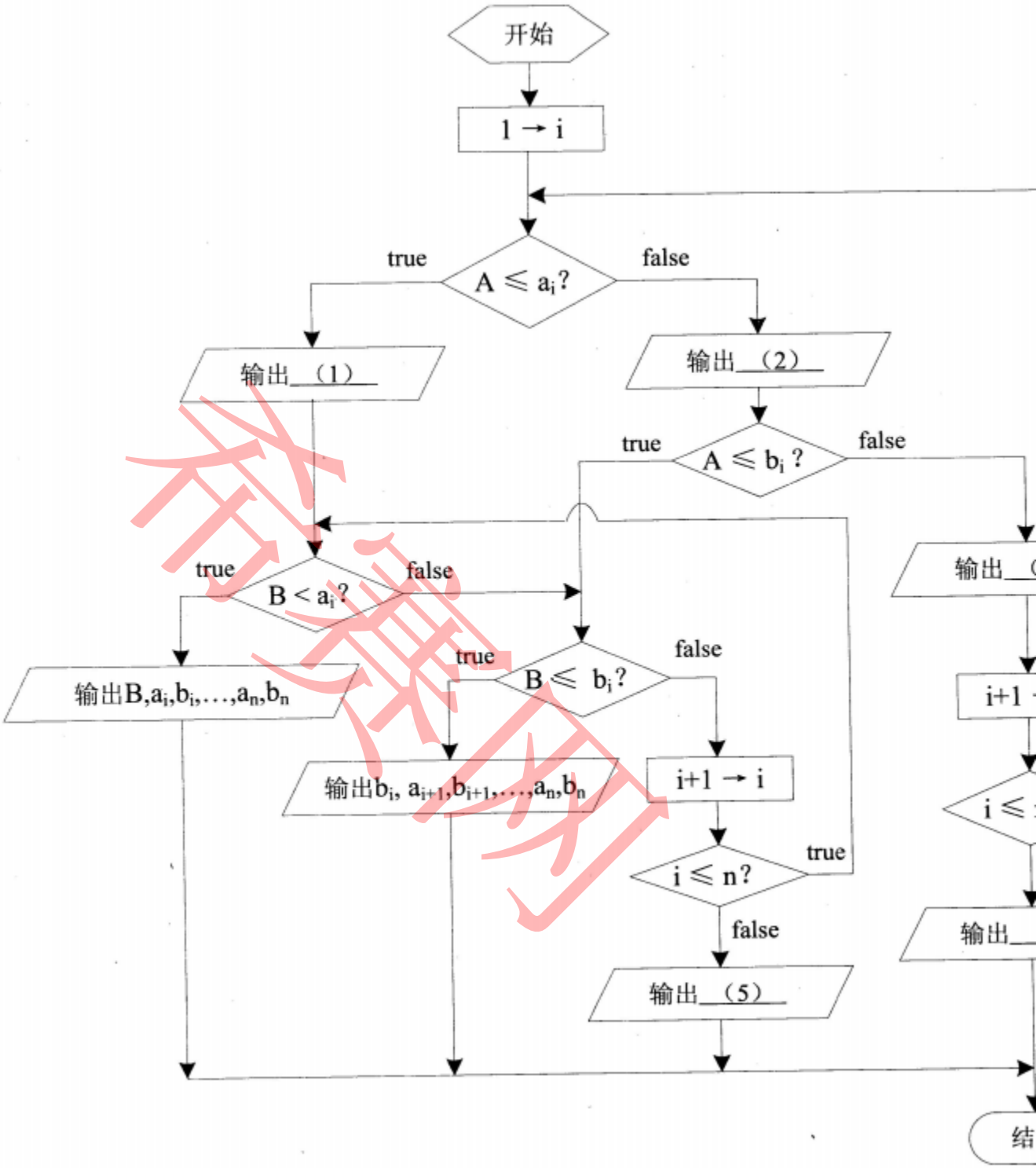
● 阅读以下说明和流程图，填写流程图中的空缺，将解答填入答题纸的对应栏内。

说明

设$[a_1b_1],[a_2,b_2],...,[a_n,b_n]$是数轴上从左到右排列的 n 个互不重叠的区间($a_1<b_1<a_2<b_2...<a_n<b_n$).以下流程图将一个新的区间[A,B](A<B)添加到上述区间集，形成新的从左到右排列的若干个互不重叠的区间(若 A、B 落在原有的两个区间，则以原有区间最左端点和最右端点为基准，形成新的区间),最后依次输出这些区间的端点。

例如，给定区间集:[1,2],[4,6],[8,10],[13,15],[17,20],添加区间[5,14]后，依次输出 1,2,4,15,17,20，表示合并后的区间集:[1,2],[4,15],[17,20]。

该流程图采用的算法是:先在 $a_1,b_1,a_2,b_2,...,a_n,b_n$ 中扫描定位 A 点，再继续描定位 B 点，在扫描过程中随时输出已确定的区间的端点值。




● 阅读以下 C 代码，回答问题 1 和问题 2，将解答填入答题纸的对应栏内。

【C 代码 1】

```c
#include <stdio.h>
int main()
{
    int a, tmp, b=0;

    scanf("%d", &a);

    tmp = a<0? -a: a;

    while (tmp) {
        b = b*10 + tmp%10;
        tmp = tmp / 10;
    }

    if (a==b || -a==b)
        printf("Palindromic number.\n");

    printf("a = %d    b = %d\n", a, b);

    return 0;
}
```

【问题 1】

写出【C 代码 1】运行时分别输入-1331、795 的输出结果。

【C 代码 2】

```
#include <stdio.h>
int main()
{
        char grade;
        int points;

        for( grade = 'A'; grade< 'F'; grade++ )    {
            switch (grade) {
                    case    'A':    points = 4 ;    break;
                    case    'B':    points = 3 ;
                    case    'C':    points = 2 ;
                    case    'D':    points = 1 ;    break;
                    case    'E':
                    case    'W':    points = 0 ;
            }
            if (points > 0)
                    printf ("Passed, points = %d\n", points ) ;
            else
                    printf ("Failed\n" ) ;
        }

        return 0;
}
```

【问题 2】

写出【C 代码 2】运行时的输出结果。

● 阅读以下说明和 C 代码，填写程序中的空(1) ～(6).将解答写入答题纸的对应栏内。

【说明】

某地电价分三档:

(1)当月用电量不超过 180 度时，每度电 0.5 元:

(2)当月用电量超出 180 度但不超过 360 度的部分，每度电 0.55 元:

(3)当月用电量超过 360 度的部分，每度电 0.7 元。

例如，某户 A 一个月的用电量为 150 度，其电费为 150*0.5=75.00 元；某户 B 用电量为 280 度，其电费为 180*0.5+(280-180)*0.55=145.00 元；某户 C 用电量为 450 度，其电费为 180*0.5+(360-180)*0.55+(450-360)*0.7=90.0+99.0+63.0=252.00 元

下面程序运行时读入 m(m>0)个住户某月的用电量，计算该月每户应缴的电费并输出，同时找出这 m 个住户中该月的最大用电量和最小用电量。

【C 代码】

```
#include <stdio.h>
#define   MAXQT   100000        //用电量的最大值

double proc(int qt)
{//计算并返回月用电量为 qt 时的电费
    double fee = 0.0;

    if ( ____(1)____ )
        fee = qt*0.5;
    else
        if ( ____(2)____ )
            fee = 180 * 0.5 + (qt - 180) * 0.55;
        else
            fee = ____(3)____;
    return fee;
}

int main()
{
    int m;                                          //住户数
    int qt, minimum=MAXQT, maximum=0;    //用电量，最小用电量，最
```

```
    scanf("%d", &m);

    while(m>0) {
        scanf("%d", &qt);
        if (qt<0 || qt > MAXQT) continue;

        printf("%.2lf\n", proc(qt));
        if ( ____(4)____ )
            minimum = qt;
        else if ( ____(5)____ )
            maximum = qt;

        ____(6)____;
    }
    printf("maximum = %d,  minimum = %d\n", maximum, minimum);
    return 0;
}
```



● 阅读以下说明和 C 代码，填写程序中的空(1) ～(6)，将解答写入答题纸的对应栏内。

【说明】

函数 insertElem 的功能是在元素升序排列的数组中加入一个新元素并保持数组元素升序排列的特点。在 main 函数中输入若干表示价格的实数，输入为 0 或负数或实数个数超出限定数量时终止，调用 insertElem 将价格按升序保存在数组 pdata 中，最后输出所输入的实数。

【C 代码】

```
#define   ARRSZ   10001
void insertElem(double arr[], int n, double elem )
/*arr 空间足够大且其元素按照升序排列，将 elem 插入 arr 中并保持其升序特点*/
{     int i;
      double tmp;

      if (n==0 || elem >= arr[n-1] ) { ____(1)____ = elem;   return; }

      for(i = n-1; i>=0 && elem < arr[i]; i--) {   //查找插入位置并将元素后移
            ____(2)____;
      }

      ____(3)____ = elem;              //将元素放入最终位置
}

int main()
{     int idx, n = 0;
      double price, pdata[ARRSZ];

      do {
          scanf("%lf", &price);
          if ( price <= 0 )   ____(4)____;
          insertElem(____(5)____); //调用 insertElem 将 price 的值加入 pdata 数组
          n++;
      }while(n<ARRSZ);

      for( idx = 0; ____(6)____; idx++)   //按升序输出所输入的实数
         printf("%.2lf\t", pdata[idx] );
      return 0;
}
```

● 阅读以下说明和 Java 程序，填写程序中的空(1) ～(5),将解答写入答题纸的对应栏内。

【说明】

以下 Java 代码实现一个简单乐器系统，音乐类(Music)可以使用各类乐器(Instrument)进行演奏和调音等操作。对部分乐器进行建模，其类图如图 5-1 所示，包括:乐器 Instrument)、打击乐器

(Percussion)、弦乐器(Stringed)、管乐器(Wind)、木管乐器(Woodwind)、铜管乐器(Brass)。

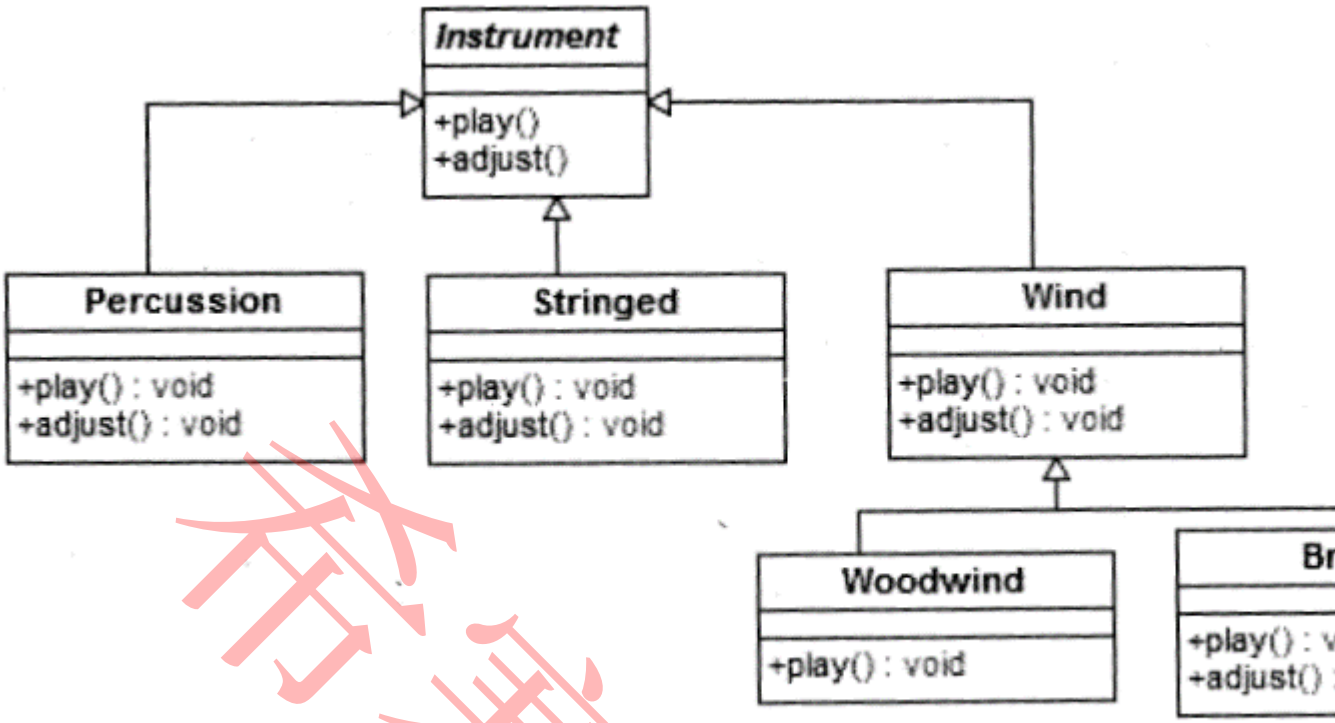



图 5-1 类图



【Java 代码】

```
import java.util.ArrayList;
enum Note{/*枚举各种音调*/
MIDDLE_C,C_SHARP,B_FLAT;        //其它略
}
abstract class Instrument{/*乐器*/
(1);                    //play 方法
abstract void adjust__(5)__    //adjust 抽象方法
}
```

```
class Wind ____(2)____{
    public void play(Note n) { System.out.println("Wind.play() " + n);   }
    public void adjust() { System.out.println("Wind.adjust()"); }
}

/* 类 Percussion 和 Stringed 实现代码略 */

class Brass ____(3)____{
    public void play(Note n) { System.out.println( "Brass.play() " + n);   }
    public void adjust() { System.out.println("Brass.adjust()"); }
}

class Woodwind extends Wind {
    public void play(Note n) { System.out.println( "Woodwind.play() " + n);   }
}

public class Music {
    void tune(Instrument i) { i.play(Note.MIDDLE_C);   }
    void adjust(Instrument i) { i.adjust();   }
    void tuneAll( ____(4)____ e ) {
        for(int j = 0; j < e.size(); j++) {
            Instrument i = e.get(j);
            adjust(i);
            tune(i);
        }
    }
    public static void main(String[] args) {
        ____(5)____ music = new Music();
        ArrayList<Instrument> orchestra = new ArrayList<>();
        orchestra.add(new Wind());
        orchestra.add(new Woodwind());
        music.tuneAll(orchestra);
    }
}
```

● 阅读下列说明和 C++代码，填写程序中的空(1) ～(5)，将解答写入答题纸的对应栏内。

【说明】

以下 C++代码实现一个简单乐器系统，音乐类(Music)可以使用各类乐器(Instument)进行演奏和调音等操作。对部分乐器进行建模，其类图如图 6-1 所示，包括:乐器(Instrument)、打击乐器(Perussion)、 弦乐器(Stringed)、 管乐器( Wind)、木管乐器(Woodwind)、铜管乐器(Brass )。

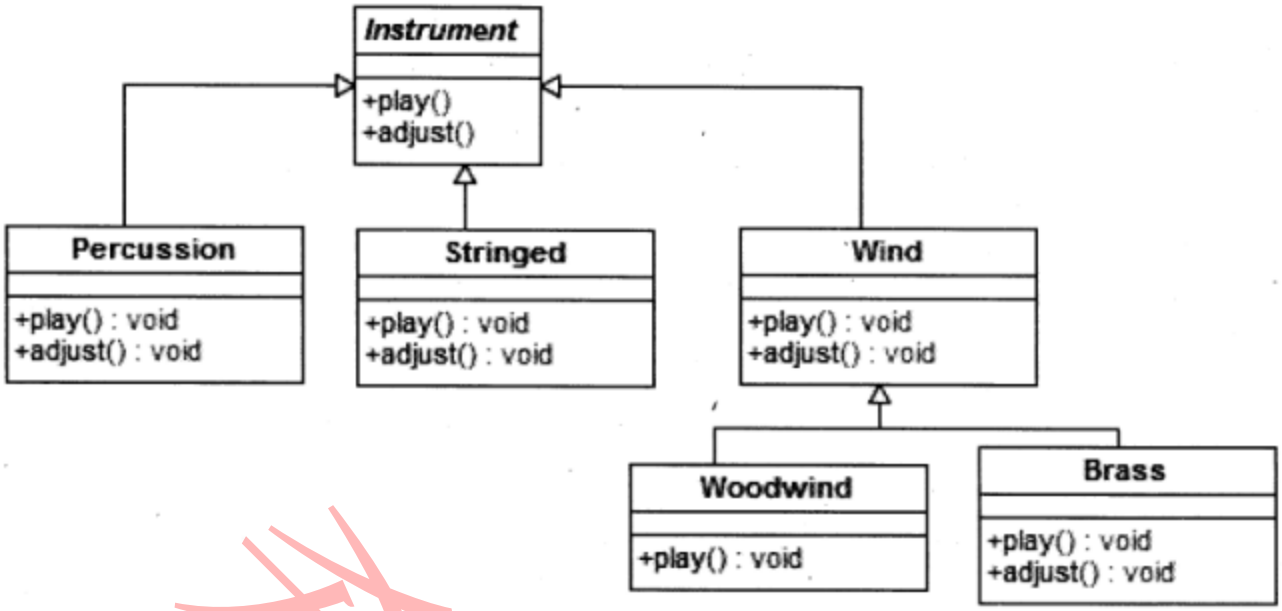



图 6-1 类图



【C++代码】

```
#include <iostream>
#include <vector>
using namespace std;
enum Note { /* 枚举各种音调 */
    MIDDLE_C, C_SHARP, B_FLAT        //其它略
};
class Instrument {/* 抽象基类，乐器 */
public:
    ___(1)___;                       //纯虚函数 play
    virtual void adjust() = 0;       //adjust 函数接口
};

class Wind :___(2)___{
public:
```

```
    void play(Note n) { cout<<"Wind.play() "<< n << endl;      }
    void adjust() { cout<<"Wind.adjust()"<<endl;      }
};

/*类 Percussion 和 Stringed 实现代码略 */

class Brass :    (3)    {
public:
    void play(Note n) { cout<<"Brass.play() " << n << endl;     }
    void adjust() { cout<<"Brass.adjust()"<<endl; }
};

class Woodwind : public Wind {
public:
    void play(Note n) { cout<<"Woodwind.play() " << n <<endl;      }
};

class Music {
public:
    void tune(Instrument* i) {      i->play(MIDDLE_C);    }
    void adjust(Instrument* i) {          i->adjust();      }
    void tuneAll(    (4)    v) {                /* 为每个乐器定调 */
        vector<Instrument*>::iterator it;
        for( it = v.begin(); it != v.end(); it++) {
            this->adjust(*it);
            this->tune(*it);
        }
    }
};

int main() {
    (5)  music = new Music();
    vector<Instrument*> orchestra;
    orchestra.push_back(new Wind());
    orchestra.push_back(new Woodwind());
    music->tuneAll(orchestra);
}
```